AD A047318

.ONICA CALIF F/G 9/2
T GUIDEBOOK: SOFTWARE QUALITY ASS--ETC(U)
F19628-76-C-0236
ESD-TR-77-255 NL

1 OF 1
ADA047318

END
DATE
FILMED
1-78
DDC

ESD-TR-77-255

12 NU

SOFTWARE ACQUISITION MANAGEMENT GUIDEBOOKS: SOFTWARE QUALITY ASSURANCE

System Development Corporation
2500 Colorado Avenue
Santa Monica, CA 90406

August 1977

Approved for Public Release;
Distribution Unlimited.

Prepared for

DEPUTY FOR COMMAND AND MANAGEMENT SYSTEMS
ELECTRONIC SYSTEMS DIVISION
HANSCOM AIR FORCE BASE, MA 01731

LEGAL NOTICE

OTHER NOTICES

This technical report has been reviewed and is approved for publication.

William J. White

WILLIAM J. WHITE, Capt, USAF
Project Engineer

John C. Mott-Smith

JOHN MOTT-SMITH, GS-13
Project Leader

John T. Holland

JOHN T. HOLLAND, Lt Col, USAF
Techniques Engineering Division

FOR THE COMMANDER

Frank J. Emma

FRANK J. EMMA, Colonel, USAF
Computer Systems Engineering Directorate

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

19 **REPORT DOCUMENTATION PAGE**

READ INSTRUCTIONS BEFORE COMPLETING FORM

18 1. REPORT NUMBER
ESD-TR-77-255

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

6 4. TITLE (*and Subtitle*)
SOFTWARE ACQUISITION MANAGEMENT GUIDEBOOK:
SOFTWARE QUALITY ASSURANCE,

5. TYPE OF REPORT & PERIOD COVERED
9 Technical rept.
Contract - Jun 1976-Aug 1977

6. PERFORMING ORG. REPORT NUMBER
14 SDC-TM-5772/001/03

10 7. AUTHOR(s)
GEORGE/NEIL
HARVEY I./GOLD

8. CONTRACT OR GRANT NUMBER(s)
15 F19628-76-C-0236

9. PERFORMING ORGANIZATION NAME AND ADDRESS
SYSTEM DEVELOPMENT CORPORATION
2500 COLORADO AVE.,
SANTA MONICA, CALIFORNIA 90406

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
PE64740F, Project 2238 16

11. CONTROLLING OFFICE NAME AND ADDRESS
Deputy for Command & Management Systems
Electronic Systems Division
Hanscom AFB, MASS 01731

12. REPORT DATE
11 AUG[redacted] 1977

13. NUMBER OF PAGES
95 12 96p.

14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)

15. SECURITY CLASS. (*of this report*)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Approved for Public Release; Distribution Unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)
SOFTWARE QUALITY ASSURANCE
SOFTWARE ACQUISITION MANAGEMENT
AIR FORCE QUALITY ASSURANCE PROGRAM
CONTRACTOR QUALITY ASSURANCE REQUIREMENTS
SOFTWARE QUALITY FACTORS

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

This report is one of a series of Software Acquisition Management Guidebooks which provide information and guidance for ESD Program Office personnel who are charged with planning and managing the acquisition of command, control, and communications system software procured under Air Force 800 series regulations and related software acquisition management concepts. It provides guidance for establishing and implementing a software quality assurance program which it discusses in terms of Program Office quality assurance requirements ~~(as defined by AFR 74-1 and ESDM 74-1)~~, contractor quality assurance requirements as defined

next page

339 900

mt

20. (cont'd)

by MIL-S-52779(AD), and software quality assurance at ESD. Special attention is given to: (1) the relationship of quality assurance to other acquisition management disciplines; (2) the integration of quality assurance requirements into the system acquisition process; (3) the contractual aspects of quality assurance; (4) monitoring the implementation of quality assurance requirements; and (5) common problems and proposed solutions.

PREFACE

This report was prepared by System Development Corporation under the direction of the Computer Systems Engineering Directorate of the Electronic Systems Division, Air Force Systems Command. The Software Quality Assurance Guidebook is one of a series of Software Acquisition Management Guidebooks intended to help ESD Program Office personnel in the acquisition of embedded software for command, control and communications systems. The contents of the guidebooks will be revised periodically to reflect changes in software acquisition policies and practices as well as feedback from guidebook users.

The Software Acquisition Management Guidebook series is currently planned to cover the following topics (National Technical Information Service accession numbers for those already published are shown in parentheses):

Regulations, Specifications and Standards (AD-A016401)

Contracting for Software Acquisition (AD-A020444)

Monitoring and Reporting Software Development Status (AD-A016488)

Statement of Work Preparation (AD-A035924)

Reviews and Audits

Configuration Management

Computer Program Development Specification (Requirements Specification)

Software Documentation Requirements (AD-A027051)

Verification

Validation and Certification

Overview of the SAM Guidebooks

Software Maintenance

Software Quality Assurance

Software Cost Estimation and Measurement

Software Development and Maintenance Facilities (AD-A038234)

Life Cycle Events (AD-A037115)

TABLE OF CONTENTS

CONTENTS (cont'd)

LIST OF FIGURES

Figure | Page

SECTION 1 - INTRODUCTION

1.1 PURPOSE

The Software Quality Assurance guidebook is designed to assist Air Force Electronic Systems Division Program Office personnel in establishing and implementing a software quality assurance program for command, control, and communications system software procured under Air Force 800-series regulations and related software acquisition management concepts. Although the discussion provided herein is intended to provide guidance for the acquisition of large-scale systems, much of it is applicable to smaller, less complex systems. However, in all cases, the guidance provided by this guidebook should be tailored to the needs of individual programs. The information provided is directed towards Program Office management personnel having quality assurance responsibility and a member of the Engineering Division, referred to as the Software Director, who is generally responsible for managing software acquisition.

1.2 SCOPE

The potential scope of quality assurance (QA), as defined in AFR 74-1, is essentially unlimited: "A planned and systematic pattern of all actions necessary to provide adequate confidence that material, data, supplies, and services conform to established technical requirements and achieve satisfactory performance." The entire concept of software acquisition management is concerned with the development of quality software. Figure 1 depicts the major PO disciplines, all of which contribute to the acquisition of quality software. In addition, Figure 1 relates each discipline to the other guidebooks in this series or to the sections within this guidebook which describe the responsibilities of each discipline.

To avoid duplication of effort with other acquisition management responsibilities, i.e., engineering management, configuration management, test management, and data management, this guidebook presents software QA in terms of:

- Program Office (PO) QA requirements as defined by AFR 74-1 and ESDM 74-1.
- Contractor QA requirements as defined by MIL-S-52779(AD).
- Software QA at the Electronic Systems Division (ESD).

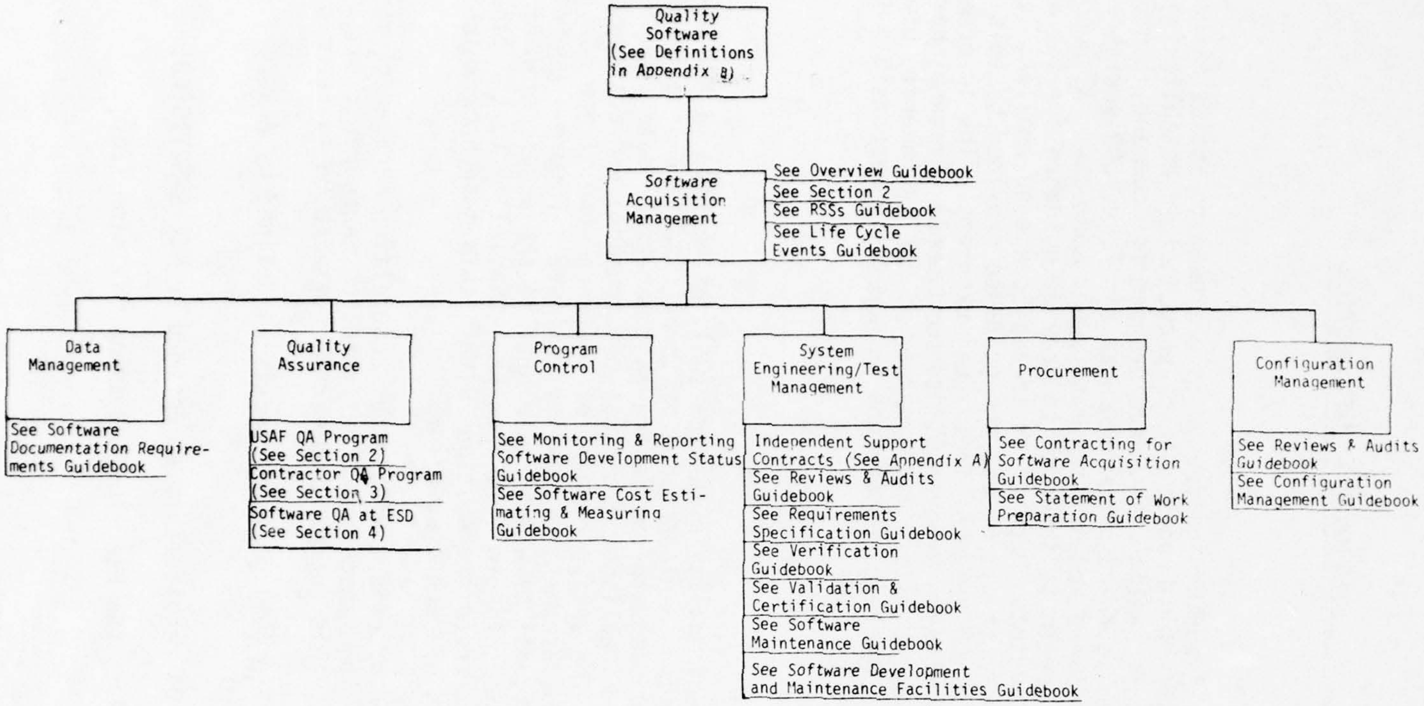

Quality Software (See Definitions in Appendix B)
Software Acquisition Management
See Overview Guidebook
See Section 2
See RSSs Guidebook
See Life Cycle Events Guidebook
Data Management
See Software Documentation Requirements Guidebook
Quality Assurance
USAF QA Program (See Section 2)
Contractor QA Program (See Section 3)
Software QA at ESD (See Section 4)
Program Control
See Monitoring & Reporting Software Development Status Guidebook
See Software Cost Estimating & Measuring Guidebook
System Engineering/Test Management
Independent Support Contracts (See Appendix A)
See Reviews & Audits Guidebook
See Requirements Specification Guidebook
See Verification Guidebook
See Validation & Certification Guidebook
See Software Maintenance Guidebook
See Software Development and Maintenance Facilities Guidebook
Procurement
See Contracting for Software Acquisition Guidebook
See Statement of Work Preparation Guidebook
Configuration Management
See Reviews & Audits Guidebook
See Configuration Management Guidebook

Figure 1. Quality Software and the Guidebook Series

Special attention in this guidebook has been given to the following:

- The relationship of QA to the other acquisition management disciplines.
- The integration of QA requirements within the system acquisition process.
- Contractual aspects of QA.
- Monitoring the implementation of QA requirements.
- Common problems and proposed solutions.
- Pitfalls, risk areas, and danger signals as they occur during the System Acquisition Life Cycle.

This guidebook identifies and describes QA activities throughout the System Acquisition Life Cycle and highlights those activities associated with software acquisition

1.2.1 Program Office QA

The PO determines the type and extent of Government QA actions required based upon the particular procurement. These actions may include:

- Inspection of products and services;
- Review of contractor's inspection/review system, quality program, or of any other means employed by the contractor to control quality and to comply with contract requirements;
- Maintenance of Government records to reflect actions, deficiencies and corrective measures.

1.2.2 Contractor QA

The contractor is responsible for controlling product quality and for offering to the Government for acceptance only those supplies and services that conform to contract requirements. The control of quality by the contractor relates to those practices, procedures, and controls employed by the contractor to assure contractual conformance.

1.2.3 Software QA at ESD

The Director of Computer Systems Engineering (MCI) is responsible for providing software support to the POs. MCI computer system personnel are assigned to the POs to assure that quality software is being developed by the responsible organizations.

1.3 CONTENTS

The subsequent contents of this guidebook are organized into three sections and three appendixes, as follows:

- Section 2 - Air Force Quality Assurance Program. Relates the Air Force QA program to the major milestones of the system acquisition cycle as they occur during the Conceptual, Validation, and Full-Scale Development Phases. Treats objectives, activities, and QA considerations for each phase. Discussions are supplemented by flow charts depicting major activities within each phase.

- Section 3 - Contractor Software Quality Assurance Programs. Provides discussions, designed to assist the PO in evaluating a contractor's proposal and the status of his software QA program. Discusses software QA responsibilities, necessary activities conducted prior to award of Full-Scale Development Phase contract, and contractor QA program implementation.

- Section 4 - Software QA at ESD. Describes how ESD assists its POs in meeting their QA requirements. Covers the evolving QA role within ESD and discusses specific QA aids.

- Appendix A - Software Quality Issues. Defines software quality and addresses the subjects of quality software vs software QA, the magnitude of QA required, and independent support contractors.

- Appendix B - Glossary. Defines (1) the major terms used in this guidebook, (2) terms related to the subject of quality software, and (3) acronyms and abbreviations used in this guidebook.

- Appendix C - Bibliography. Lists books, papers, and military regulations, specifications, and standards that are particularly relevant to the subject of this guidebook.

SECTION 2 - PROGRAM OFFICE QA REQUIREMENTS

2.1 INTRODUCTION

AFR 74-1 is the primary regulation governing QA. It is complemented by AFR 74-18 and ESDM 74-1. These are the official documents used to establish the basic QA requirements within this section.

The three primary software-related QA objectives of the PO are to assure that:

- The technical and contractual requirements for the CPCI(s), data, and services are practical and enforceable.

- The delivered CPCI(s), data, and services conform to the specified technical and contractual requirements.

- The causes of user dissatisfaction and mission degradation are identified and corrected or eliminated.

These PO QA objectives are derived from the basic requirements of the Air Force QA program. Within this context, the PO is responsible for assuring that these requirements are clearly identified and that responsibility for the satisfaction of each requirement is clearly assigned to one of the organizations participating in the system development effort. The PO's QA organization supports the definition of system QA requirements and assures that the responsible organizations meet their assigned requirements. Thus, to be meaningful and effective, the implementation of the Air Force QA program must involve all PO and contractor organizations. The PO's QA organization must coordinate the total QA effort.

The remainder of this section shows the relationship between software QA and other PO activities during the Conceptual (2.2), Validation (2 3), and Full-Scale Development (2.4) Phases. A series of flow charts (Figures 2, 3. and 4) depicts the relative sequence of major technical and management milestones during each phase, which is then discussed in terms of QA objectives, technical and management activities, quality review of end products, and common problems with proposed solutions.

2.2 CONCEPTUAL PHASE

2.2.1 QA Objectives During the Conceptual Phase

The basic objective of the Conceptual Phase is to define the system requirements to the level of a System Specification and to establish plans for the acquisition and control of the system during the System Acquisition Life Cycle. QA activities during this phase are aimed at establishing an appropriate quality program and reviewing the Conceptual Phase products. The following six major documents are developed during this phase:

- Program Management Directive (PMD)
- Program Management Plan (PMP)
- Decision Coordination Paper (DCP)
- System Specification (SS)
- Test & Evaluation Master Plan (TEMP)
- Validation Phase Request for Proposal (RFP)

Of these documents, the System Specification impacts most heavily upon software quality. However, the other documents are also important since they establish direction for management, testing, cost, and scheduling and it is often difficult to make major changes of direction in these areas during succeeding phases.

2.2.2 Conceptual Phase Activities

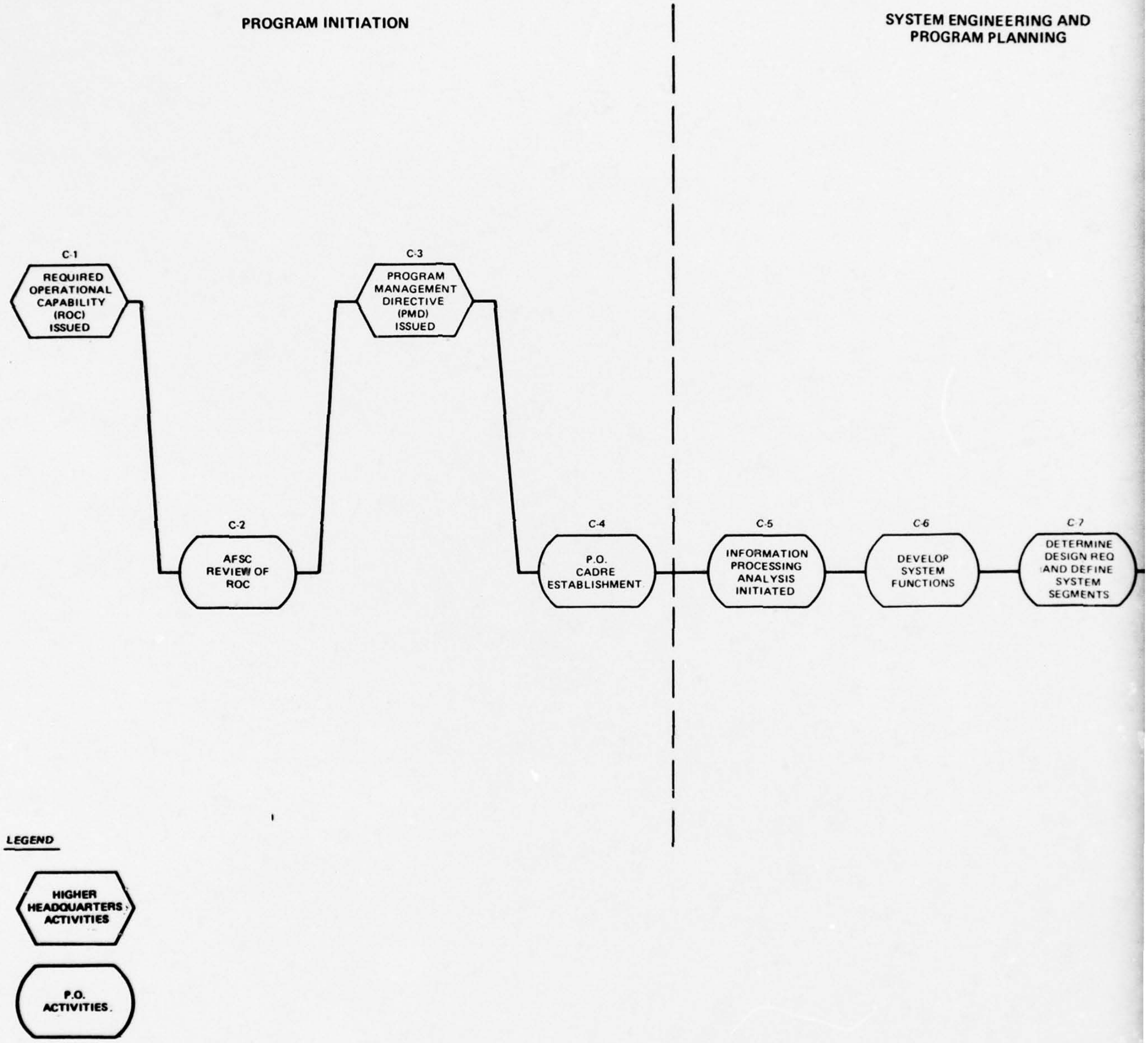
The major activities of the Conceptual Phase are as follows:

- Program initiation
- System engineering and program planning
- Document system requirements and prepare RFP.

Figure 2 illustrates the typical sequence of these activities. For each package of activities a summary description is provided in the following paragraphs along with a discussion of the QA considerations.

2.2.2.1 Program Initiation

Program initiation is devoted to evaluating the proposed new operational capability to determine its feasibility and to establishing a PO for managing the system acquisition. The associated Conceptual Phase milestones (see Figure 2) are:


PROGRAM INITIATION
SYSTEM ENGINEERING AND PROGRAM PLANNING
C-1
REQUIRED OPERATIONAL CAPABILITY (ROC) ISSUED
C-2
AFSC REVIEW OF ROC
C-3
PROGRAM MANAGEMENT DIRECTIVE (PMD) ISSUED
C-4
P.O. CADRE ESTABLISHMENT
C-5
INFORMATION PROCESSING ANALYSIS INITIATED
C-6
DEVELOP SYSTEM FUNCTIONS
C-7
DETERMINE DESIGN REQ AND DEFINE SYSTEM SEGMENTS
LEGEND
HIGHER HEADQUARTERS ACTIVITIES
P.O. ACTIVITIES

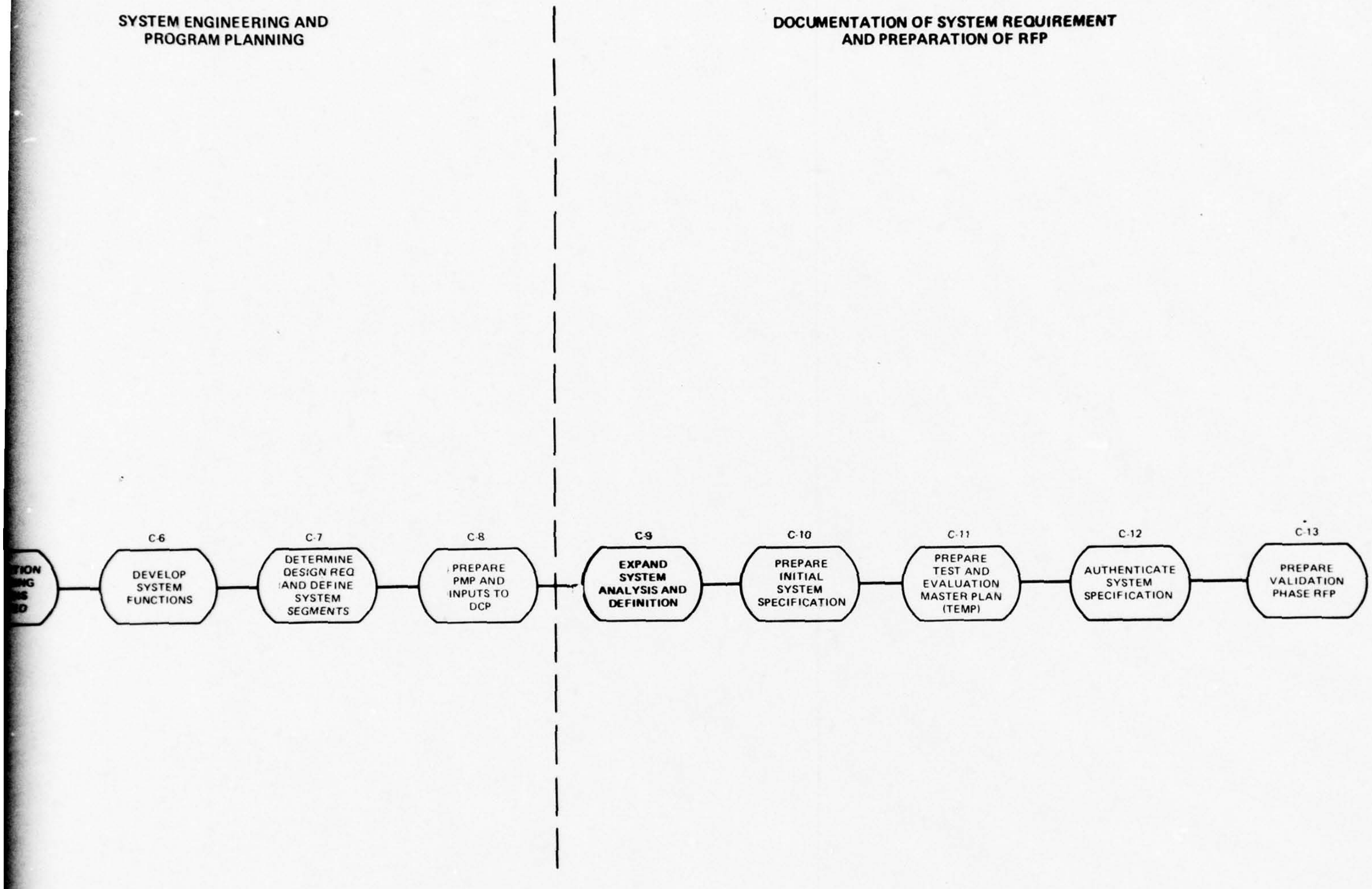

SYSTEM ENGINEERING AND PROGRAM PLANNING
DOCUMENTATION OF SYSTEM REQUIREMENT AND PREPARATION OF RFP
TION NG S D
C-6
DEVELOP SYSTEM FUNCTIONS
C-7
DETERMINE DESIGN REQ AND DEFINE SYSTEM SEGMENTS
C-8
PREPARE PMP AND INPUTS TO DCP
C-9
EXPAND SYSTEM ANALYSIS AND DEFINITION
C-10
PREPARE INITIAL SYSTEM SPECIFICATION
C-11
PREPARE TEST AND EVALUATION MASTER PLAN (TEMP)
C-12
AUTHENTICATE SYSTEM SPECIFICATION
C-13
PREPARE VALIDATION PHASE RFP

Figure 2. Conceptual Phase Process

- Required Operational Capability (ROC) Issued - BLOCK C-1
- AFSC Review of ROC - BLOCK C-2
- Program Management Directive (PMD) Issued - BLOCK C-3
- PO Cadre Established - BLOCK C-4

2.2.2.1.1 Summary of Activities

The ROC identifies the need for a new or improved operational capability. Once the ROC is validated by HQs USAF, the PMD, which authorizes AFSC to establish a Program Office Cadre, is issued.

2.2.2.1.2 QA Activities

The QA activities, during this early stage of the System Acquisition Life Cycle, amount to reviewing the PMD to gain an understanding of the program objectives and the management direction provided by HQ USAF. This information provides the basis for initiating QA planning activities. (See Task 1, Review of Program Management Directive, ESDM 74-1.) However, since the PMD is issued before the PO is established, the QA activities do not include a quality review of the PMD.

AFR 800-14, Vol. II, Chapter 3, provides guidance for computer resources planning, including data that should be included in the PMD. Attention should be directed to paragraph 3-1 which states, "This Guidance applies to the case in which the computer resources are known to be required at the outset". With many systems this information is just not available until a substantial amount of system engineering has been accomplished. Paragraph 3-6 contains specifics regarding computer-resource information within the PMP. AFR 800-14, Vol. II, does not specify whether information regarding computer resources is available with the initial PMD at the beginning of the Conceptual Phase or with the updated PMD issued near the end of the Conceptual Phase. The level of information regarding computer resources will be determined on a system by system basis.

2.2.2.2 System Engineering and Program Planning

System engineering and program planning are the first activities to be performed by the PO. These activities include the initial system engineering required to define the total system requirements and the initiation of the system planning activities. The associated blocks in the Conceptual Phase flow (see Figure 2) are:

- Information Processing Analysis Initiated - BLOCK C-5

- Develop System Functions - BLOCK C-6
- Determine Design Requirements and Define System Segments - BLOCK C-7
- Prepare PMP and Inputs to DCP - BLOCK C-8

2.2.2.2.1 Summary of Activities

Conceptual Phase system engineering activities are concerned with defining the system in functional terms and establishing requirements for the total system. Procurement strategy will be established for the follow-on phases. (See Contracting for Software Acquisition guidebook, page 15, for a discussion of the Advanced Procurement Plan). This strategy must be considered in the allocation of requirements to system segments. The main purpose of segmenting the system is to define packages which can be offered for competitive bidding by industry. The planning activities involve the total system during all phases of the System Acquisition Life Cycle. The technical definition arrived at during these activities should be sufficient to provide the technical base for preparing:

- The PMP
- Inputs to the DCP
- The System Specification

2.2.2.2.2 QA Activities

The three areas of software QA concern during system engineering and program planning are:

- QA program as defined within the PMP.
- The definition of system requirements and the allocation of the requirements to segments.
- DCP inputs.

The QA Program As Defined Within the PMP. The PMP is developed by the PO to document the plan for managing the system acquisition. Information regarding the contents and preparation of a PMP is provided in AFSCP 800-3, Attachments 3 and 4. Procedures for review of the PMP, and a model PMP, are presented in ESDM 74-1, Section 4-2. Section 4.2.3 of ESDM 74-1 is applicable to any system and generally applicable to any type of Configuration Item (CI).

The following paragraphs in the Model PMP should be modified to incorporate software quality requirements:

- Section 3, paragraph a, should be modified as follows: The PO procurement quality assurance program will be in accordance with AFR 74-1 and AFSCR 74-6. PCOs will require in the contract that contractors establish, implement, and maintain a quality program in accordance with MIL-Q-9858A, MIL-C-45662A, and other quality assurance documents that may be specified. For Computer Program CIs, the quality program will be conducted in accordance with MIL-S-52779(AD).

- A new statement should be added to Section 4-1, as follows: Assurance that the software supplies and services are in conformance with contractual requirements will be determined by the contractor under the cognizance of the CAO. The technical evaluation of the contractor products will be accomplished by the PO.

In reviewing the PMP to determine the adequacy of planning information relative to the contemplated software developments, the PO should ascertain if:

- The system engineering requirements are adequate, realistic, and compatible with the estimated software developmental requirements.

- The master schedule reflects software development considerations and is compatible with initial or revised software development estimates.

- The initial planning reflects considerations for development and support facilities for software. (See Software Development and Maintenance Facilities guidebook.)

- The initial planning includes software transfer and turnover requirements.

- Provisions have been made to develop a Computer Resources Integrated Support Plan (CRISP). (See AFR 800-14, Vol. II, page 3-4, paragraph 3-8.)

- A Validation Phase has been scheduled. If not, provisions have been made to accomplish the activities leading up to the generation of the CPCI Development (Part I) Specification(s).

The Definition of System Requirements and Their Allocation to Segments. When a prime contractor is to be awarded total system responsibility, segmenting of the System Specification is not required. However, when segmentation is required it should be accomplished prior to entering into the Validation Phase, and the segments should be identified within the System Specification/Segment Specifications. To properly review the segmentation of a system, the system segment concept, as defined by the Air Force, must be understood. This concept is not adequately treated in currently available regulations, specifications, and standards.

System programs generally require the participation of several industrial companies and Government agencies. An important function of the system segment concept is to apportion system requ'rements into packages which will eventually be assigned to the participating organizations. The segment structure must be determined before the Validation Phase RFPs are issued, even though the structure may later be changed. The system segments contain allocated requirements from the System Specification along with their functional interface definitions. These segments are used by Validation Phase contractors as parametric limits within which CIs in each segment will be identified and specified. These limits are used as boundries of potential Full-Scale Development contracts expressed in functional terms. The central idea in the system segment concept is to define the contractual relationships between the Government and contractor, not to constrain technical thought. Each segment will be the responsibility of one and only one contractor or Government agency.

DCP Inputs. The DCP basically documents the current status of the system, identifies requirements for follow-on development, and documents a formal request to proceed with the system. For small programs (something less than a major system) a Program Memorandum may be substituted for a DCP. (See AFSCP 800-3, page 1-2, paragraph 1-6.)

DCP inputs from the PO should be reviewed to determine if a Validation Phase is being proposed. One of the primary objectives of the Validation Phase is to prepare a Development Specification for each CI. Without a Validation Phase it is very difficult to allocate enough time and effort to adequately prepare these specifications. The CPCI Development Specification is the single most important document contributing to the development of quality software. Without a good CPCI Development Specification, the PO and contractors are placed in a situation of inventing or discovering performance requirements during the Full-Scale Development Phase. Consequently, risk is greatly increased and the probability of completion within projected costs and schedules is greatly decreased.

2.2.2.3 Document System Requirements and Prepare RFP

The concluding Conceptual Phase activities include the following milestones (see Figure 2):

- Expand System Analysis & Definition - BLOCK C-9
- Prepare Initial System Specification - BLOCK C-10
- Prepare Test and Evaluation Master Plan (TEMP) - BLOCK C-11
- Authenticate System Specification - BLOCK C-12
- Prepare Validation Phase RFP - BLOCK C-13

2.2.2.3.1 Summary of Activities

The final activities during the Conceptual Phase consist of documenting the results of the technical and planning activities and preparing the RFP for the Validation Phase. The initial System Specification is prepared, authenticated,* and baselined.**

2.2.2.3.2 QA Activities. A quality review should be conducted on the end products of the Conceptual Phase and will include the following:

- System Specification
- Test and Evaluation Master Plan
- Validation Phase RFP
- Advanced Procurement Plan

*Authenticate - Approval signature by a responsible person of the procuring activity. Authentication by the procuring activity normally will be accomplished on that issue of the specification which is to be the contractual requirement for the baseline which that particular specification defines [MIL-STD-483(USAF) paragraph 3.4.9].

**AFSCM/AFLCM 375-7, Chapter 2, paragraph 2.7.

System Specification. ESDM 74-1, Task 5, provides general information regarding quality review of specifications. Note that this information clearly identifies the difference in responsibilities between the Engineering Division and the QA organization, i.e., the Engineering Division is responsible for the technical contents of the specification whereas the QA personnel are responsible for reviewing it in terms of insuring clarity, preventing misinterpretation, and avoiding ambiguity. The System Specification becomes the contractual baseline for the Validation Phase contracts. The purpose of the specification is to state the Government requirements in a way that will be intelligible to all potential contractors and to the Government representatives who must administer the contract after it is issued. Hence, the ideal specification is one that is so clear and definite that it is not subject to interpretation during performance. The quality review is aimed at approaching the ideal specification by removing redundant, conflicting, and unclear statements of requirements, thus helping to avoid future contractual and technical problems.

Since an initial System Specification is almost always incomplete and obviously not ideal, there is often reluctance to baseline it at the end of the Conceptual Phase. However, in the interest of good management and for the benefit of both the PO and the contractor(s), there must be a baselined specification to control performance, costs, and schedules. Changes to the specification are inevitable, but each change should be thoroughly reviewed for impact on costs and schedules. Without a baselined specification, control of performance, costs, and schedules becomes very difficult.

The adequacy of the System Specification has a major impact on the system acquisition because the System Specification is the basis for future planning, reviews, and milestones. The following specific items within the System Specification should be reviewed for QA considerations:

- The software-related system segments must be reviewed to determine if the requirements are clearly stated and contain sufficient detail to initiate the CI definition process. The functional interface definition must clearly establish the boundaries of each segment. If the functional interfaces are not complete, problems regarding areas of responsibility are likely to occur during the Validation Phase. The impact of incomplete interface definitions must be evaluated on a case-by-case basis, depending on the particular procurement situation, e.g., associate contractors competing for segments or prime contractors competing for the total system development.

- The system requirements must be sufficiently detailed; they must be feasible and enforceable. Feasibility and enforceability judgements are PO system engineering responsibilities which should be backed up by documented system engineering studies. --Appendix 1, MIL-STD-490, and Appendix III, MIL-STD-483(USAF) provide guidance on the contents of the System Specification.

- Any design constraints must be reasonable and necessary. The design constraints should be absolutely required and not just desirable, because they impose limitations on the ways the performance requirements can be implemented, and consequently may prohibit lower cost implementation methods. Design constraints should not conflict with any performance requirements. For example, whenever a specific computer or computer configuration is required, there must be an engineering analysis which demonstrates that the performance requirements can be met within the imposed limitations.

- The system capacities and accuracies must be defined. --System capacities refer to capacities for the total system, e.g., maximum number of intercepts, maximum tracks, maximum number of sensors. This information is critical in detailing the requirements for the application software.

Note

Task 5 (Specification Review) of ESDM 74-1 points out that, "quality personnel should not question the engineering requirements or design, but will assure incorporation of adequate controls."

Test and Evaluation Master Plan (TEMP). The TEMP should be reviewed for software test planning considerations. Based upon current knowledge of software performance requirements, and of system implementation schedules, particular attention should be given to feasibility of test plans and compatibility of test schedules, for example:

- Software testing cannot take place until the required software, facilities, equipment, and personnel are available.

- Requirements for provision of test inputs and for analysis of test outputs must be compatible with plans and schedules for generating input data and to support recording, reduction, and analysis of output data.

Validation Phase RFP. The heart of the RFP is the statement of work (SOW) which provides a description of the tasks to be accomplished during the contract period. AFSCP 800-6, Chapter 2, provides general guidance on preparing a SOW. Paragraph 2-6 provides an SOW checklist which can be used when reviewing an SOW. Chapter 6 of AFSCP 800-6 provides specific instruction regarding preparation of an SOW for a Validation Phase contract. Attention should be given to the advice in paragraph 2-4,g; "Do not overspecify. The ideal situation is to specify results required or the end items to be delivered and let the selected contractors find the best methods of getting there. In any case, he should not be told exactly how to do it and then be made responsible for the results. To support the most favorable type of contract, describe clearly and fully what is required to satisfy the contract".

Because of problems that have been experienced in software acquisitions there is a tendency to overspecify, particularly in the area of design constraints and controls. Caution should be exercised when determining the proper level of detail. In certain areas, when the Government specifies a level of detail, they must accept responsibility down to that level. The Government should not be placed in the position of accepting responsibility prematurely. The Validation Phase SOW should call for the contractor to prepare a QA Plan as part of the Full-Scale Development Phase Proposal [see MIL-S-52779(AD)].

A Full-Scale Development specimen SOW is prepared by the PO and released with the Validation Phase RFP. This specimen SOW provides guidance for contractors to prepare their Full-Scale Development Phase proposals. For further specific software SOW preparation guidance see the Statement of Work Preparation guidebook and ESDM 74-1.

Advanced Procurement Plan. The single most important document in software acquisition is the CPCI Development (Part I) Specification. This specification is normally generated by the contractor during the Validation Phase and becomes the contract specification for the Full-Scale Development contract. The Advanced Procurement Plan should be reviewed to determine if provisions have been made to have the Development Specification generated during the Validation Phase. If no Validation Phase is planned, then provisions should have been made, in the schedule, to generate the Development Specification at the early stages of Full-Scale Development. When this specification is generated during Full-Scale Development, the PO must be sensitive to the fact that they have entered into a contract without having an approved contract specification (CPCI Development Specification). The plan should recognize this and provide for establishing the Development Specification as the contract baseline and making Full-Scale Development contract adjustments if necessary.

2.2.3 Common Conceptual Phase QA Problems and Proposed Solutions

Many system problems are not realized until the system is being developed. However, the cause of these problems, in most cases, can be traced to inadequate technical planning during the Conceptual Phase. Such problems include:

- System Specifications are ambiguous and do not clearly define requirements.

- Definition of system segments and allocation of requirements to segments is inadequate.

- Failure to provide for system/software engineering studies which support the feasibility of the performance requirements to be established during the Validation Phase.

- A decision to speed up the development process by bypassing the Validation Phase and delaying generation of Development (Part I) Specifications until the Full-Scale Development Phase.

- SOWs are ambiguous and not clearly defined.

- Insufficient attention is given to ensuring that initial software requirements are compatible with development schedules and budgets.

- Deviations from AFR 800 series acquisition policy and procedures are made without evaluating the impact on contracting and technical development.

- Insufficient time, effort, and experience is devoted by the PO or system engineering contractor to developing a sound System Specification.

- Confusing and contradictory RSSs that assume the reader has extensive experience in systems acquisition.

- System and software engineering processes are not adequately defined within the RSSs.

- Need for more training programs for PO personnel with emphasis on Air Force acquisition procedures, system engineering, and system planning.

- Severe schedule constraints imposed by externally enforced schedules.

2.2.3.1 Proposed Solutions

The Conceptual Phase deals with the total system and does not focus upon the software. Problems created during this phase tend to be system problems. Therefore, the solutions should be looked at from the system point of view as follows:

- Selection of PO personnel and system engineering contractors should be based on successful experience in related activities and systems.

- Each inexperienced member of a PO should attend an extensive course on the acquisition and management of systems. Special sessions should be given on acquisition and management of embedded software.

- Advice regarding acquisition policies and procedures should be solicited from knowledgeable personnel in the Acquisition Support Office or Computer Systems Engineering Office (ESD/DR or ESD/MCI).

- The Air Force acquisition process contains a set of unique terms that communicate requirements and direction. Avoid introducing semantic problems which are likely to confuse the reading audience. "Inconsistency and ambiguity are general problems of the software industry, so in the context of acquisition it is doubly important that definitions be precise to avoid (1) technical misinterpretation and (2) contractual difficulties."*

- Define and plan the system as a total entity and not as a group of individual elements, e.g., hardware, software, communications.

- Insist on a sound system specification as the baseline for the Validation Phase.

2.3 VALIDATION PHASE

The objective of the Validation Phase is to validate the choice of performance alternatives and to provide a sound basis for determining whether to preceed with the Full-Scale Development Phase. For computer software, the major product of this phase is the contractor's proposal for the Full-Scale Development Phase. This proposal includes the contractor's CPDP and the Development (Part I) Specifications and test plans for each CPCI. During this phase, the PO updates the initial System Specification and prepares the CRISP.

Since the introduction of the Validation Phase in the System Acquisition Life Cycle, many problems and constraints have been introduced into the system development process, mainly because of the lack of guidance on how to conduct a Validation Phase. Most definitions of the Validation Phase emphasize hardware proofing and prototype demonstrations which are basically applicable to hardware. AFSCP 800-6, Chapter 6, states that three groups of tasks may be included in the Validation Phase, i.e., (1) systems and program definitions, (2) hardware proofing, and (3) prototype demonstration.

*Quoted from Software Documentation Requirements guidebook.

For software, the emphasis is on system and program definitions. This set of tasks refers to the refinement and definition of the System Specification to a lower level of performance requirements (Allocated Baseline). These efforts may also include modeling and coding of performance-sensitive areas and may be performed by competing contractors. Quality assurance requirements should be included in the SOW written by the PO. During the Validation Phase, the contractor develops the required program plans, and submits a proposal for the Full-Scale Development Phase. As part of the source selection activities to pick the Validation Phase contractor(s), pre-award surveys should be conducted. The surveys should include an inspection of the internal procedures and controls proposed by the potential contractors for controlling their software development activities. These internal procedures include such areas as:

- Configuration Management
- Error reporting
- Quality Assurance
- Documentation
- Management reviews and reports

2.3.1 Validation Phase Activities

Figure 3 depicts the typical sequence of software activities during a Validation Phase. These activities are primarily system engineering tasks rather than software design tasks. Software designers are used to support system engineering. Basically the software designers investigate the design feasibility of the stated performance requirements.

The success of any program is largely determined by how well the Conceptual and Validation Phase activities were performed. The outputs of the Validation Phase provide the technical requirements and management plans which form the basis for establishing the contractual agreements for developing and controlling the CPCIs during the Full-Scale Development Phase.

The activities of the Validation Phase are grouped into the following three major packages:

- System segment analysis
- CPCI requirements definition
- Completion of Validation Phase Products

The technical evaluation of the products is the responsibility of the Engineering Division. The QA organization's basic job is to verify that everything assigned has been accomplished. In the following paragraphs, a summary description of each package of activities is provided along with a discussion of the related QA considerations.

2.3.1.1 System Segment Analysis

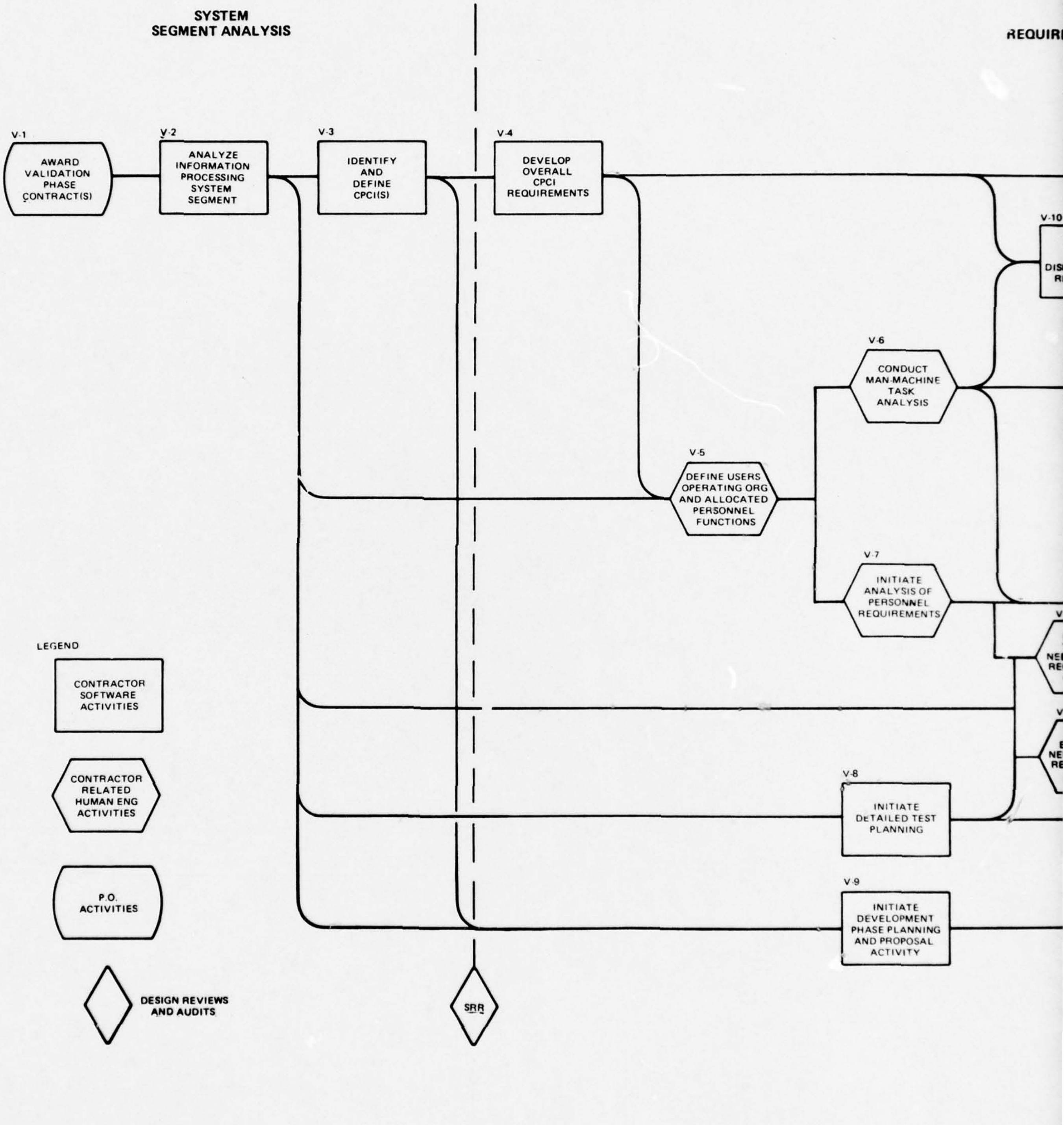
The initial contractor activity during the Validation Phase is the development of the initial CPCI definitions. This activity is terminated by a System Requirements Review (SRR). The associated blocks in the Validation Phase flow chart (see Figure 3) are:

- Award Validation Phase Contract(s) - BLOCK V-1
- Analyze Information Processing System Segment(s) - BLOCK V-2
- Identify and define CPCI(s) - BLOCK V-3

2.3.1.1.1 Summary of Activities

System segment analysis is concerned with (1) expanding the detail of system functions allocated to the information processing-related system segment, (2) detailing associated performance/design requirements, and (3) determining suitable implementation methods. These activities are another iteration of the process which began during the Conceptual Phase and which will continue through the Full-Scale Development Phase. Each iteration derives the additional level of detail required to proceed to the next step of system design and development, i.e., develop the level of detail necessary for allocating the implementation of design requirements among CPCIs, manual operations, or joint man/machine operation. The packaging of functions into CPCIs has a major impact on the success of the system development effort. Therefore, it is essential that the contractor and the SD mutually understand the rationale for defining and selecting CIs. A hardware or computer program CI, by definition, must meet the following criteria:

- It is a physical and functional part of a system.
- It represents the contractor's highest level of assembly for delivery and the PO's unit of management.
- It is a common reference for engineering task descriptions, contracts, schedules, and budgets.
- Each developmental CI will be specified in a separate Development (Part I) Specification and a separate Product (Part II) Specification.


SYSTEM
SEGMENT ANALYSIS
REQUIR
V-1
AWARD VALIDATION PHASE CONTRACT(S)
V-2
ANALYZE INFORMATION PROCESSING SYSTEM SEGMENT
V-3
IDENTIFY AND DEFINE CPCI(S)
V-4
DEVELOP OVERALL CPCI REQUIREMENTS
V-10
V-5
DEFINE USERS OPERATING ORG AND ALLOCATED PERSONNEL FUNCTIONS
V-6
CONDUCT MAN-MACHINE TASK ANALYSIS
V-7
INITIATE ANALYSIS OF PERSONNEL REQUIREMENTS
V-8
INITIATE DETAILED TEST PLANNING
V-9
INITIATE DEVELOPMENT PHASE PLANNING AND PROPOSAL ACTIVITY
SRR
LEGEND
CONTRACTOR SOFTWARE ACTIVITIES
CONTRACTOR RELATED HUMAN ENG ACTIVITIES
P.O. ACTIVITIES
DESIGN REVIEWS AND AUDITS

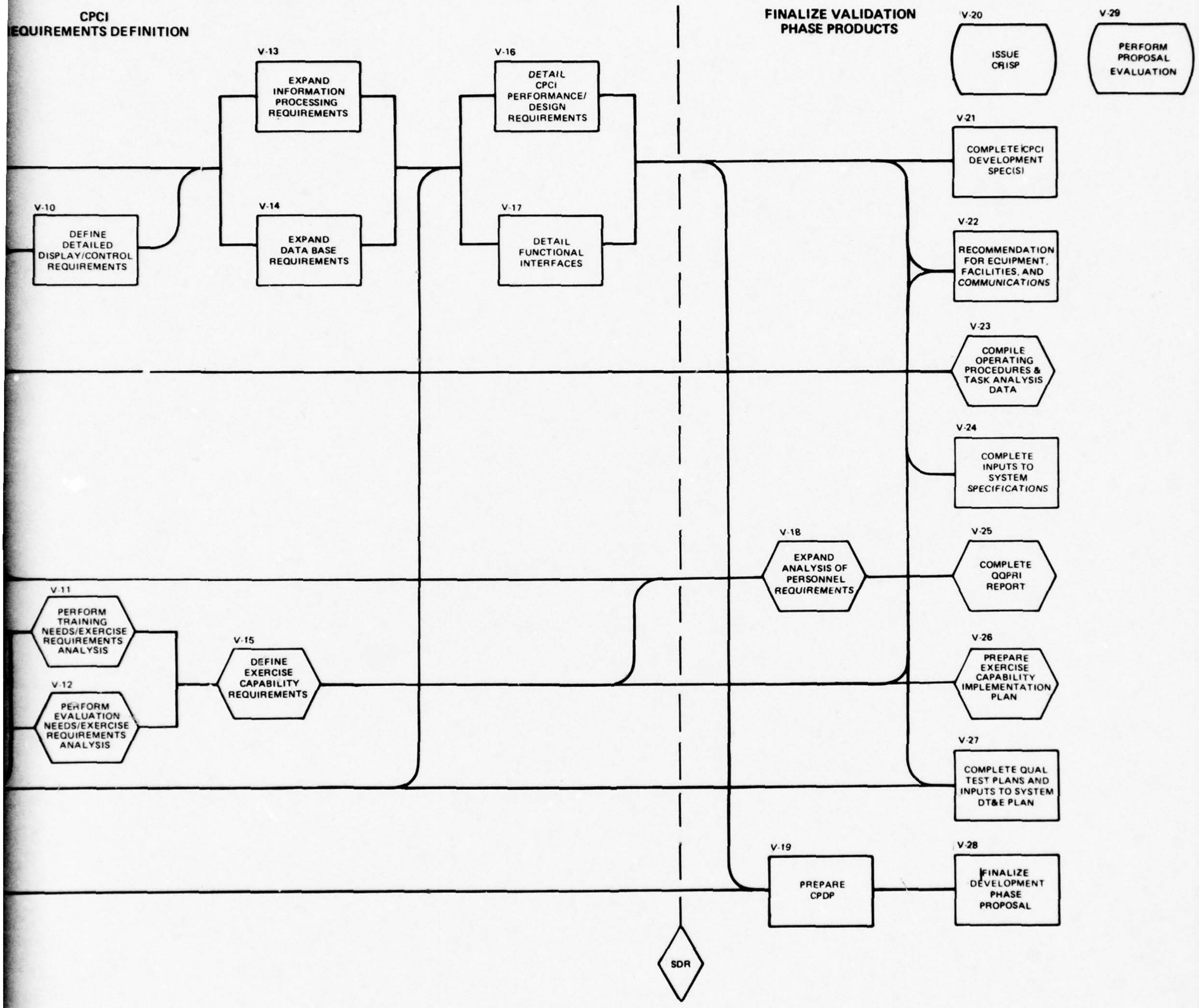

CPCI
EQUIREMENTS DEFINITION
FINALIZE VALIDATION PHASE PRODUCTS
V-10 DEFINE DETAILED DISPLAY/CONTROL REQUIREMENTS
V-11 PERFORM TRAINING NEEDS/EXERCISE REQUIREMENTS ANALYSIS
V-12 PERFORM EVALUATION NEEDS/EXERCISE REQUIREMENTS ANALYSIS
V-13 EXPAND INFORMATION PROCESSING REQUIREMENTS
V-14 EXPAND DATA BASE REQUIREMENTS
V-15 DEFINE EXERCISE CAPABILITY REQUIREMENTS
V-16 DETAIL CPCI PERFORMANCE/ DESIGN REQUIREMENTS
V-17 DETAIL FUNCTIONAL INTERFACES
V-18 EXPAND ANALYSIS OF PERSONNEL REQUIREMENTS
V-19 PREPARE CPDP
V-20 ISSUE CRISP
V-21 COMPLETE CPCI DEVELOPMENT SPEC(S)
V-22 RECOMMENDATION FOR EQUIPMENT, FACILITIES, AND COMMUNICATIONS
V-23 COMPILE OPERATING PROCEDURES & TASK ANALYSIS DATA
V-24 COMPLETE INPUTS TO SYSTEM SPECIFICATIONS
V-25 COMPLETE QQPRI REPORT
V-26 PREPARE EXERCISE CAPABILITY IMPLEMENTATION PLAN
V-27 COMPLETE QUAL TEST PLANS AND INPUTS TO SYSTEM DT&E PLAN
V-28 FINALIZE DEVELOPMENT PHASE PROPOSAL
V-29 PERFORM PROPOSAL EVALUATION
SDR

Figure 3. Validation Phase Process

2

- It represents the lowest level of management control by the PO.

- It is the item to be qualified, delivered, and accepted.

- It is the item to be placed under configuration management control.

Although CI definition is primarily a technical task, it must be tempered by the needs of other system development requirements. (See Section 2 of the Configuration Management guidebook.)

2.3.1.1.2 QA Activities

The System Requirements Review (SRR) is the first Validation Phase milestone which allows the PO to evaluate the developer's progress. The SRR is primarily a technical review conducted by the Engineering Division. QA personnel participate to monitor the accomplishment of the review. The SRR is an in-process review and its scheduling at this stage in the requirements definition process is critical. The intent of the SRR is to evaluate the developer's progress and the direction of the initial Validation Phase effort. It is conducted at the system/segment level.

The technical description of activities conducted during the SRR is discussed in the following SAM Guidebooks:

- Verification

- Validation and Certification

- Reviews and Audits

The primary QA concerns are to assure that the following determinations have been made:

- The requirements stated in the System Specification are the point of departure for all future system development activities. It is critical that all participating organizations interpret these requirements in a consistent manner so that compatibility during development can be attained.

- The SOW for the Validation Phase should have identified specific tradeoff studies to be accomplished by the developer. The PO should determine to what extent these studies have been accomplished. They may not necessarily be complete by SRR but must be complete by SDR.

- The system segment interfaces define the boundaries of the areas of development responsibilities. The development organizations must recognize these interfaces and their relationship to the other segments and systems. At the SRR a check should be made with the developer to determine if the interfaces are understood and are being reflected in the requirements definition activities.

- The contractor should show which requirements in the System Specification or Segment Specification have been allocated to specific CPCIs.

- The initial definition of the CPCIs must be scrutinized very closely as it has a major impact on the remainder of the Validation Phase and more importantly on the Full-Scale Development Phase contracts, technical activities, and deliverables.

The instruction found in Section 2 of the Computer Program Configuration Management guidebook identifies the criteria used for selecting CPCIs. The areas impacted by CPCI selection, in addition to the technical impact, are cost, schedule, configuration management, interface control, documentation, testing, integration, and contracting. Care should be taken to ensure that the CPCI selection is evaluated in terms of all areas of impact.

2.3.1.2 CPCI Requirements Definition

CPCI requirements definition consists of (1) detailing the performance requirements for each identified CPCI, (2) preparing test planning information, (3) conducting man-machine analysis, and (4) initiating Full-Scale Development Phase planning activities. CPCI requirements definition activities are terminated by a System Design Review (SDR). The associated blocks in the Validation Phase flow chart (see Figure 3) are:

- Develop Overall CPCI Requirements - BLOCK V-4.

- Define Users Operating Organization and Allocated Personnel Functions - BLOCK V-5.

- Conduct Man-Machine Task Analysis - BLOCK V-6.

- Initiate Analysis of Personnel Requirements - BLOCK V-7.

- Initiate Detailed Test Planning - BLOCK V-8.

- Initiate Development Phase Planning & Proposal Activity - BLOCK V-9.

- Define Detailed Display/Control Requirements - BLOCK V-10.

- Perform Training Needs/Exercise Requirements Analysis - BLOCK V-11.
- Perform Evaluation Needs/Exercise Requirements Analysis - BLOCK V-12.
- Expand Information Processing Requirements - BLOCK V-13.
- Expand Data Base Requirements - BLOCK V-14.
- Define Exercise Capability Requirements - BLOCK V-15.
- Detail CPCI Performance/Design Requirements - BLOCK V-16.
- Detail Functional Interfaces - BLOCK V-17.

2.3.1.2.1 Summary of Activities

Based upon identification of major functions to be performed by the CPCIs, CPCI requirements definition undertakes a further analysis and breakdown of the functions into sub-functions and tasks.

For each CPCI function/subfunction, it is necessary to identify the source and form of input data, initial operations performed, all logical manipulations and computations to be accomplished, and the relevant outputs/interfaces with other functions, together with alternative modes of operations, and rules of operation. Although this is basically a system engineering task, support is provided by computer program designers who will be developing an initial design to determine if the performance requirements are feasible.

In detailing personnel requirements, the identified functions to be performed by operational personnel will fall into the following two broad categories:

- Manual. Manual functions are those which do not imply direct interaction with the computer, but which are essential to the operational mission. These functions will include decisions, planning, coordinating, communication, status posting, etc. Manual functions are also those performed by command and/or staff personnel of the operational organization.
- Man-Machine. Man-Machine functions are those which are directly associated with computer operation, to be performed by personnel at, or having access to, consoles, displays, or other input/output equipment.

In addition to detailing performance requirements, the test planning activity is initiated at this time and continues throughout the System Acquisition Life Cycle. This activity represents a progressive refinement and expansion of the material initially contained in the TEMP. The TEMP, prepared by the PO, is the basic guidance document for the test activity. It contains overall test philosophy, basic concepts and objectives for Subsystem DT&E and System DT&E, and rudimentary test planning information.

For detailed test planning, it is necessary to expand the basic test concepts and objectives to:

- Incorporate the developer's particular approach to developing his CPCI(s).
- Integrate the test concepts and objectives into the development approach.
- Develop preliminary CPCI DT&E planning information.

2.3.1.2.2 QA Activities

QA during CPCI requirements definition is performed in conjunction with the SDR. The SDR is a system engineering review conducted before the developer finalizes the Validation Phase products. A detailed discussion of the SDR can be found in the Reviews and Audits guidebook. QA items to be considered include:

- When draft CPCI Specifications are available at the SDR, the principal QA function at the SDR is to determine the compatibility between Sections 3 and 4 of the specifications. For information regarding the contents of Section 4 of a CPCI Development (Part I) Specification, see the Computer Program Development Specification guidebook. Section 4 of the specification should normally contain a Verification cross reference matrix which indicates the method to be used to testing each performance requirement stated in Section 3. The matrix should be supported by the text within Section 4. There is a tendancy to have planning information in Section 4. This should be avoided and Section 4 should be restricted to requirements information only.
- Evaluate the definition of the CPCIs using the criteria stated in Section 2 of the Computer Program Configuration Management guidebook.

- Verify that traceability exists between the requirements in the System Specification, system segments, and CPCI requirements.

- Review the CPCI requirements to determine if they are stated clearly and unambiguously. This will avoid contractual and technical problems during the Full-Scale Development Phase.

- Determine the reasons for "not applicable" and "to be determined" sections in the specifications. Review the schedule for completing the "to be determined" sections.

- Evaluate test planning information, in particular the CPCI DT&E Plan to determine:

  - If it is compatible with the requirements and schedules in the TEMP.
  - If it addresses all requirements stated in Section 4 of the associated CPCI Development (Part I) Specification.

2.3.1.3 Complete Validation Phase Products

These final tasks in the Validation Phase are conducted by the contractor(s) for the purpose of completing his deliverable products. These products basically constitute his proposal for the Full-Scale Development Phase contract. The associated blocks on the Validation Phase flow chart (see Figure 3) are:

- Expand Analysis of Personnel Resources - BLOCK V-18.

- Prepare CPDP - BLOCK V-19.

- Issue CRISP - BLOCK V-20.

- Complete CPCI Development Specifications - BLOCK V-21.

- Complete recommendations for equipment, facilities*, and communications - BLOCK V-22.

- Compile Operating Procedures and Task Analysis Data - BLOCK V-23.

- Complete inputs to System Specification - Block V-24.

- Complete QQPRI Report - BLOCK V-25.

*See Software Development and Maintenance Facilities guidebook.

- Prepare Exercise Capability Implementation Plan (see DI-H-3270A) - BLOCK V-26.

- Complete Qualification Test Plans (CPCI Cat I Test Plan) and Inputs to System DT&E Plan - BLOCK V-27.

- Finalize Development Phase Proposal - BLOCK V-28.

- Perform Proposal Evaluation - BLOCK V-29.

With the exception of Blocks V-20 and V-29, all milestones are performed by the contractor. Task 6, Source Selection/Technical Evaluation, ESDM 74-1, should be used in conjunction with Block V-29, Perform Proposal Evaluation.

2.3.1.3.1 Summary of Activities

The Computer Resources Working Group (CRWG) is responsible for preparing, updating, and issuing the CRISP. The CRISP identifies organizational relationships and responsibilities for the management and technical support of computer resources. Responsibilities for computer resources are normally allocated to the development command, using command, and support command. The detailed contents of the CRISP are identified in AFR 800-14, Volume II, paragraph 3-8.

The contractor's chief activity at this time is the preparation of his final report. Typically it will contain the following types of information:

1.0 Introduction and Brief Summary

2.0 Technical Report

2.1 Trade Study Conclusions

2.1.1 Man-Machine Functional Allocations

2.1.2 Development of Mathematical Equations

2.1.3 Alternate Information Processing Flows

2.1.4 Display Design

2.1.5 Manual Input Alternatives

2.1.6 Timing and Storage Requirements

2.2 System Engineering Documentation

2.2.1 Functional Allocation

a. Computer Program Functions

b. Operator Functions

2.2.2 Operator Task Analysis

2.2.3 Built-in Simulation and Test Capabilities

2.2.4 Data Reduction Capabilities

2.2.5 Required Computer Program Development Tools

2.2.6 Personnel Requirements Information for Operational, Computer Program Support, and (when applicable) Simulation/System Exercising Personnel

2.2.7 Recommended Design Requirements for Equipment, Communications, and Facilities

2.3 System Specification Expansion

2.3.1 Definition and List of CPCIs

2.3.2 Functional Allocations

2.4 CPCI Development Specification(s) [MIL-STD-483(USAF), Appendix VI]

2.4.1 Operational Requirements

2.4.2 Support Requirements

2.4.3 Utility Requirements

2.5 CDRL for Full-Scale Development Phase

3.0 Contractor's Development Phase Program Management Plans

3.1 Typical planning areas to be covered (some of which are covered in the CPDP) include the following:

3.1.1 Organization and Personnel Management

3.1.2 Technical Management

3.1.3 Detailed Integration During the Development Phase

3.1.4 Development

3.1.5 Test

3.1.6 Installation and Checkout

3.1.7 Financial

3.1.8 Procurement

3.1.9 Personnel and Training

3.2 Additional planning activities, which may be included, are: human factors, Government-Furnished Property (GFP), PERT/cost, test-site responsibilities, configuration management, exercise capability, installation, and orientation.

2.3.1.3.2 QA Activities

QA activities at this stage of the Validation Phase are performed while evaluating the contractor's Validation Phase products which leads to selection of the Full-Scale Development Phase contractor.

The PO's technical evaluation should normally emphasize the proposed CPCI Development (Part I) Specification and the updated System Specification together with the associated system engineering documentation delivered at the end of the Validation Phase. The objective of the PO's evaluation is to determine if Validation Phase obligations have been met (by analyzing the technical adequacy and completeness of Validation Phase products). Most importantly, the PO should determine if:

- All risk items for Full-Scale Development have been identified.

- The risks have been minimized by the contractor's performance of such tasks as detailed design feasibility studies, simulations, and development of key risk software.

- The system engineering organization is confident that the CPCI Development (Part I) Specifications are complete and adequate for proceding into Full-Scale Development. Often technical review of the Development Specification and its supporting Validation Phase studies should be performed by a System Engineering/Technical Direction (SE/TD) contractor. (See Section 3 of Appendix A of this guidebook.)

To ensure that the Full-Scale Development Phase has a sound base, the following basic questions must be answered during the PO's evaluation of the Validation Phase products:

- Is it clear what the CPCI(s) must do (PERFORMANCE REQUIREMENTS)?
- How do the CPCI(s) relate to the system (INTERFACE REQUIREMENTS)?
- Does the contractor display the knowledge to perform the development activities?
- Will the proposed contract enable the contractor to perform satisfactorily and will adequate visibility be provided for the PO?
- Are the schedules reasonable and compatible?

Validation Phase products to be reviewed and evaluated are as follows:

CPCI DEVELOPMENT SPECIFICATIONS

- Are the requirements stated in performance terms with a minimum of design constraints?
- Are the stated design constraints necessary and acceptable to the PO?
- Does Section 4, "Quality Assurance Requirements", take into account all requirements identified in Section 3, "Requirements"?
- Are all requirements stated in Section 3 able to be examined or tested?
- Are there any conflicting requirements?
- Are the stated requirements feasible and have the high risk areas been eliminated?
- Are the CPCI requirements compatible and traceable to the allocated system/segment requirements?
- Are the Section 5, "Preparation for Delivery", requirements specified?*
- Are all TBDs (to be defined) justified?

*MIL-STD-483(USAF), Appendix VI, states that Section 5 of the CPCI Development Specification is normally not applicable. However, if the requirements for delivery are to be contractually binding they must be reflected in the CPCI Development Specification.

UPDATED SYSTEM SPECIFICATION*

- Have the functional interfaces been detailed?
- Is there a firm list of contractor-proposed CPCIs?
- Is there traceability between system/segment requirements and the CPCIs?
- Are there any required clarifications to the basic requirements within the System Specification?
- Are there any proposed changes to the basic system requirements?

TEST PLANNING DATA**

- Are the test planning data compatible with Section 4 of the Development Specification?
- Are the test schedules compatible with the overall test program as reflected in the TEMP?
- Are the Preliminary Qualification Tests (PQTs) adequate or are there too many?
- Does the PO have the capability to manage the proposed test program?
- Are all tests identified at performance-requirements level rather than design?
- Can System DT&E CPCI-related requirements be accomplished during Subsystem DT&E?

*At this time changes to the System Specification are normally confined to the detailing of the system segments.

**The test planning data supplied by the contractor include a Qualification Test Plan (DI-T-3703) for each CPCI and inputs to the System DT&E Plan. For further information on test planning data see the Verification and Validation & Certification guidebooks.

PROPOSED CONTRACTOR DATA REQUIREMENTS LIST*

- Does the CDRL call for the minimum data necessary?
- Has the contractor provided back-up sheets to the CDRL?
- Does the PO agree with the tailoring of the DIDs?
- Is the data packaged consistently with the CPCI definitions?
- Are the data delivery schedules consistent with the CPCI/system development schedules?

CONTRACTOR'S MANAGEMENT CONTROLS

- Configuration Management Procedures
  - Are they compatible with contractual requirements?
  - Are the internal controls adequate for processing and controlling changes?
  - Is the software engineering release system effective?
  - Are the specification and documentation maintenance procedures adequate?
  - Is the problem/error reporting system, prior to product baseline, adequate?
- Computer Program Development Plan
  - Does the contractor have a sound approach to the development of the CPCI?
  - Does the development plan provide for proper sequencing of the Air Force monitoring and control milestones, particularly PDR, CDR, PQT, FQT, and PCA?
  - Are the tasks in the CPDP reflected in the WBS so that costs can be properly identified, collected and evaluated?

*For further information regarding the CDRL and documentation selection see the Software Documentation Requirements guidebook.

- Contractor's QA program
  - No DID exists for the procurement of a contractor's QA program. The Software Documentation Requirements guidebook provides instructions for modifying the CPCP DID to meet this need.
- Cost Control
  - Is the level of cost reporting adequate for cost control and reporting of the software development effort?

2.3.2 Common Validation Phase QA Problems and Proposed Solutions

Many of the computer program problems experienced during the Full-Scale Development Phase are created as a result of inadequate Validation Phase decisions and tasks. Some of these problems include:

- CPCIs are defined in relatively small design packages in hopes of achieving visibility and control over development. In actuality what happens is:
  - The PO accepts more responsibility.
  - Data costs are increased.
  - Interface control problems increase (increasing the number of CIs and their interfaces).
  - More complex configuration management requirements are generated (more baselines).
  - More PDRs and CDRs result.
  - Qualification testing is done in smaller design packages.
  - Greater system integration problems occur after CPCI qualification.
- Unnecessary design information is included in the CPCI Development (Part I) Specification. When the Government approves and baselines a specification and the specification contains both performance requirements and design requirements, the contractor satisfies the lowest level of approved requirements. If there is a conflict between a performance statement and a design constraint the contractor need only satisfy the design constraint. When a requirement is stated in a specification and the Government approves the specification, the Government is accepting responsibility for the specification. The contractor's responsibility is to satisfy the specification.

- The inability of contractors to write good CPCI Development Specifications and the inability of PO personnel to evaluate the adequacy of CPCI Development Specifications. This, by far, is the greatest single problem in the development/acquisition of CPCIs. Many contractors will take advantage of this situation by using ECPs to further define the performance requirements, at the same time increasing costs.

- The CPDP inadequately describes the proposed development process.

2.3.2.1 Proposed Solutions

The results of the Validation Phase should provide a sound and realistic base for establishing the Full-Scale Development Phase contract. Serious mistakes made at this time make it almost impossible to have a successful development. When carefully applied, the following guidelines will minimize the problems encountered in the Full-Scale Development Phase:

- The most important document to insure quality in software development is the CPCI Development (Part I) Specification. The Development Specification states the performance requirements for the CPCI, i.e., what the CPCI will do. This specification presents "design to" requirements and, for application CPCIs, will be stated in mission operational terms, not in data processing terms. During the Full-Scale Development Phase the Air Force is contracting for CPCIs that will satisfy these performance statements. Sophisticated design approaches may be used, but if it is not clear what the CPCI is expected to do, the design will not provide the solution.

- Whenever possible, insist on a Validation Phase.

- Allow the contractor to define the CPCIs and have the PO approve the CPCI selection.

- A CPCI should be considered the largest integrated design package delivered, using the rationale for selection outlined in Section 2 of the Configuration Management guidebook. Examine all impact areas (e.g., including cost, data, configuration management, and test) prior to approving the CPCI definition.

- Remove unnecessary design constraints from the Development Specification prior to approval.

- Baseline the CPCI Development Specification at contract award, thus providing an approved point of departure for controlling changes during Full-Scale Development Phase activities.

- The PO should require detailed descriptions of the contractor's proposed development approach in the CPDP. Information should be included about programming standards, management and control practices, and reporting procedures. The CPDP should correlate fully with the contractor's proposed WBS, costs, and schedules. If not, there is little likelihood that the CPDP will be followed.

- Competent detailed technical review and evaluation must be applied to all Validation Phase products. If the PO has insufficient qualified review personnel, an SE/TD contractor should be used. (See Section 3 of Appendix A of this guidebook.)

2.4 FULL-SCALE DEVELOPMENT PHASE

The basic software objectives of the Full-Scale Development Phase are (1) to design and develop the CPCI(s) that satisfies the associated CPCI Development Specification(s), (2) to generate the related support documentation, and (3) to integrate the CPCI(s) with the other CIs in the system. The primary products of the software activities in the Full-Scale Development Phase include:

- Qualified and accepted CPCI(s).

- Updated Development (Part I) Specifications.

- Approved and baselined Product (Part II) Specifications.

- A support data package for each CPCI, including instructions for operating and using the CPCI.

- Configuration management records which provide CPCI/ECP configuration status information.

2.4.1 Full-Scale Development Phase Activities

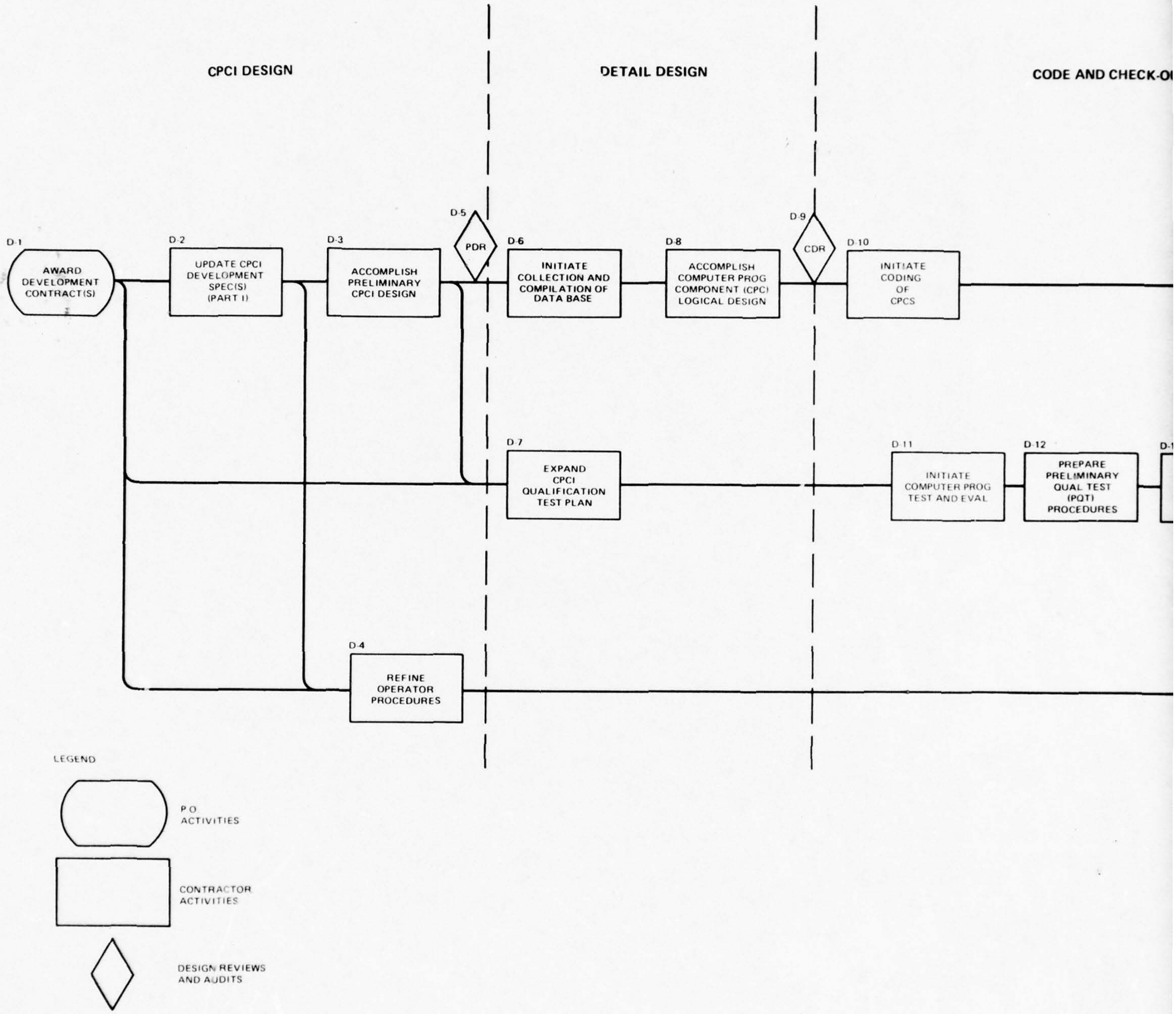
Figure 4 provides a flow chart depicting the typical sequence of CPCI events during a Full-Scale Development Phase. These activities can be grouped into four packages as follows:

- CPCI design

- Detail design

- Code and checkout

- CPCI qualification and acceptance


CPCI DESIGN
DETAIL DESIGN
CODE AND CHECK-OU
D-1
AWARD DEVELOPMENT CONTRACT(S)
D-2
UPDATE CPCI DEVELOPMENT SPEC(S) (PART I)
D-3
ACCOMPLISH PRELIMINARY CPCI DESIGN
D-5
PDR
D-6
INITIATE COLLECTION AND COMPILATION OF DATA BASE
D-8
ACCOMPLISH COMPUTER PROG COMPONENT (CPC) LOGICAL DESIGN
D-9
CDR
D-10
INITIATE CODING OF CPCS
D-7
EXPAND CPCI QUALIFICATION TEST PLAN
D-11
INITIATE COMPUTER PROG TEST AND EVAL
D-12
PREPARE PRELIMINARY QUAL TEST (PQT) PROCEDURES
D-1
D-4
REFINE OPERATOR PROCEDURES
LEGEND
P.O. ACTIVITIES
CONTRACTOR ACTIVITIES
DESIGN REVIEWS AND AUDITS

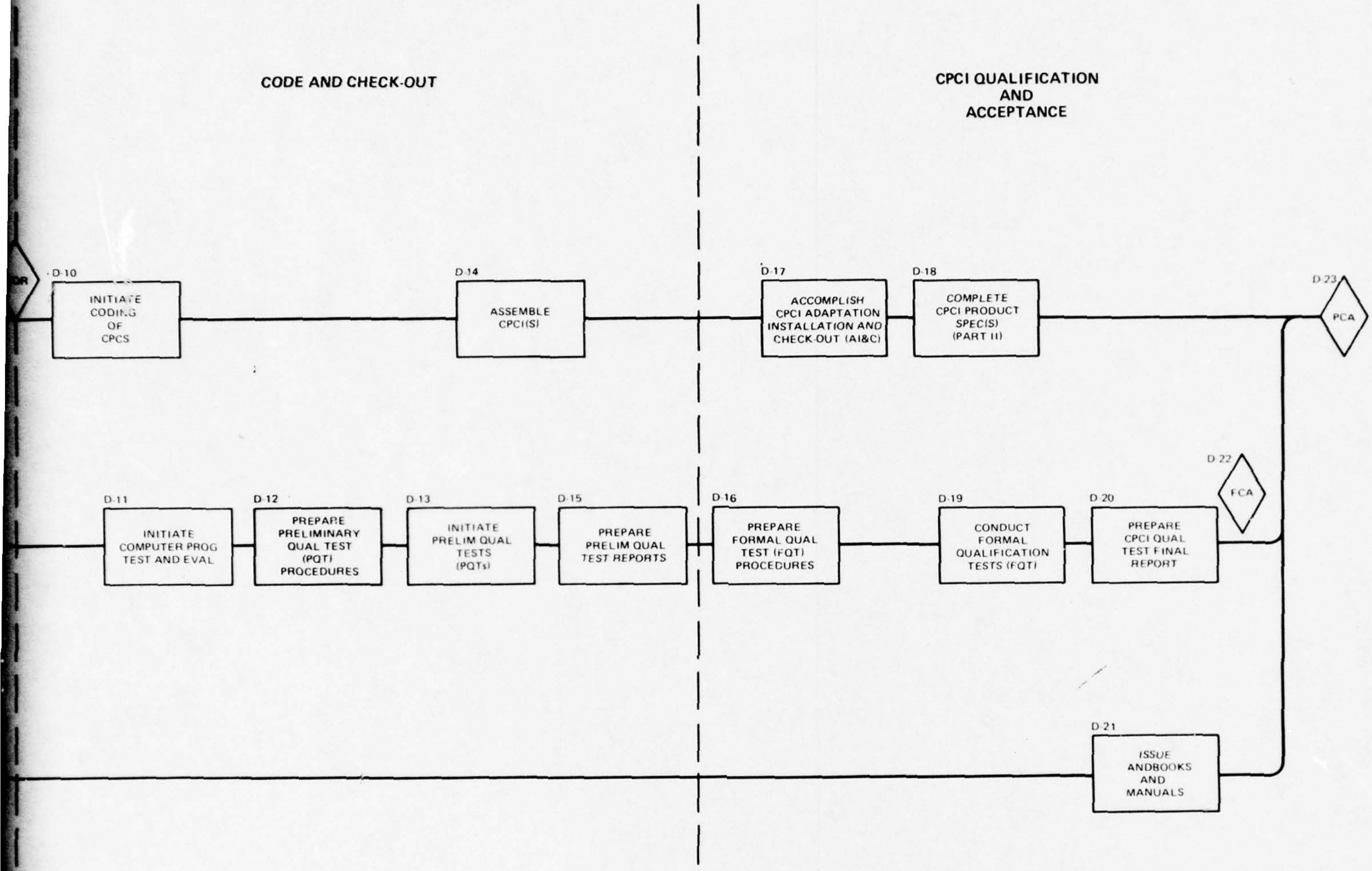

CODE AND CHECK-OUT
CPCI QUALIFICATION AND ACCEPTANCE
DR
D-10
INITIATE CODING OF CPCS
D-14
ASSEMBLE CPCI(S)
D-17
ACCOMPLISH CPCI ADAPTATION INSTALLATION AND CHECK-OUT (AI&C)
D-18
COMPLETE CPCI PRODUCT SPEC(S) (PART II)
D-23
PCA
D-22
FCA
D-11
INITIATE COMPUTER PROG TEST AND EVAL
D-12
PREPARE PRELIMINARY QUAL TEST (PQT) PROCEDURES
D-13
INITIATE PRELIM QUAL TESTS (PQTs)
D-15
PREPARE PRELIM QUAL TEST REPORTS
D-16
PREPARE FORMAL QUAL TEST (FQT) PROCEDURES
D-19
CONDUCT FORMAL QUALIFICATION TESTS (FQT)
D-20
PREPARE CPCI QUAL TEST FINAL REPORT
D-21
ISSUE ANDBOOKS AND MANUALS

Figure 4. Full-Scale Development Phase Process

2

A summary of each package of activities follows:

2.4.1.1 CPCI Design

The intent of this group of activities is to develop a design for each CPCI and review the adequacy of that design. The associated blocks in the Full-Scale Development Phase flow chart (see Figure 4) are:

- Award Development Contract - BLOCK D-1
- Update CPCI Development Specification - BLOCK D-2
- Accomplish Preliminary CPCI Design - BLOCK D-3
- Refine Operator Procedures - BLOCK D-4
- Conduct PDR - BLOCK D-5

2.4.1.1.1 Summary of Activities

The purpose of this set of activities is to (1) develop the overall CPCI design based on the requirements of the approved CPCI Development (Part I) Specification and (2) to prepare for the Preliminary Design Review (PDR). At the outset of CPCI design, the Development Phase contracts are awarded and the contractor's proposed software QA program is evaluated and approved. The CPCI Development (Part I) Specification(s) are then updated to reflect the interfaces with the selected equipment/CIs. The PDR is an engineering design review conducted on the CPCI prior to committing it to the detail design process. For further information regarding PDRs, see the Reviews and Audits guidebook. Additionally, during CPCI design, operator procedures are refined in preparation for operator/user manual development.

2.4.1.1.2 QA Activities

QA activities for CPCI design are performed in conjunction with the system engineering review at PDR and are intended to ensure that:

- The CPCI Development Specifications are updated and baselined.
- All Development Specification performance requirements are allocated within the CPCI design.
- The status of approved ECPs and their impact upon design are known.

- The Subsystem DT&E Test Plan is compatible with Section 4 of the CPCI Development Specification.
- The overall CPCI design is reviewed including:
  - A description of overall information flow
  - The structure of the CPCI data base
  - A description of the CPCI control structure
  - The identification of CPCs
  - An estimate of storage allocations and identification of critical timing areas
- All the utility and support software requirements are defined and scheduled.
- The contractor's plans for controlling changes to the design are established, understood, and followed. (See Task 8, Engineering Inspections and Reviews, ESDM 74-1.)

2.4.1.2 Detail Design

Detail design refers to the activities occurring between the PDR and the Critical Design Review (CDR). The associated blocks in the Development Phase flow chart are:

- Initiate Collection and Compilation of Data Base - BLOCK D-6
- Expand CPCI Qualification Test Plan - BLOCK D-7
- Accomplish CPC Logical Design - BLOCK D-8
- Conduct CDR - BLOCK D-9

2.4.1.2.1 Summary of Activities

The purpose of detail design is to expand the content of the CPCI design, as presented at PDR and to produce documentation describing the detailed logic to be used in developing code for individual CPCs.

The collection and compilation of the data base is also initiated at this time. This activity consists of the collection and subsequent compilation of computer program data elements obtained from sources other than the contractor responsible for computer program development. Data elements to be collected include those which describe the natural environment of the system, characteristics of weapons to be employed in the system, and other similar data.

2.4.1.2.2 QA Considerations

QA considerations for detail design are performed in conjunction with the CDR. The CDR is a formal technical review concerned with establishing the integrity of the CPCI design prior to coding and testing and is thus an engineering management responsibility. The CDR may be scheduled in increments for a complex CPCI which is scheduled to reach any given stage of the design/ development/test process in increments of CPCs. Detailed discussions of CDR activities are contained in the Reviews and Audits guidebook. Verification of CPCI design is discussed in the Verification guidebook.

Task 8, Engineering Inspections and Reviews, ESDM 74-1 describes the CDR in terms of a hardware CDR. There are some basic differences between CPCI and equipment CI CDRs that should be realized by the SD. The hardware CDR is used to support a production decision. The equipment prototype should have been developed and qualified. The CDR associated with CPCIs occurs during development, not at the end, and is basically used as a point for determining if the programmers have worked out the design prior to coding.

From a QA standpoint the SD should review:

- Traceability of detailed design to the PDR data and the CPCI Development Specification.

- Approved ECP implementation status and its impact upon design and coding.

2.4.1.3 Code and Checkout

Code and checkout refers to the contractor activities of coding computer programs and conducting internal testing [Computer Program Test and Evaluation (CPT&E)] to shake-down the design. The associated blocks in the Development Phase flow chart (see Figure 4) are:

- Initiate Coding of CPCs - BLOCK D-10

- Initiate CPT&E - BLOCK D-11

- Initiate Preliminary Qualification Test (PQT) Procedures - BLOCK D-12
- Initiate PQTs - BLOCK D-13
- Assemble CPCI(s) - BLOCK D-14
- Prepare PQT Reports - BLOCK D-15

2.4.1.3.1 Summary of Activities

The CPC logical design, presented at CDR, is now coded. As soon as coding for the first logically discrete computer program module is complete, the contractor will initiate CPT&E. The objective of CPT&E is to verify the integrity of the code with respect to its design. A second objective is to determine the ability of the various CPCs, when assembled, to communicate with each other, to operate as a unit, and to correctly process inputs and produce the required outputs. CPT&E is a subset of DT&E. It is the CPCI informal testing performed by the contractor, at his discretion, to support his design and development effort. It is of no real benefit to the PO to witness CPT&E. However, the PO should assure that CPT&E has been performed in accordance with the CPDP. Contractor development activities, such as code walk throughs, parameter tests, use of controlled test versions, and maintenance of development and test records, may be monitored or reviewed depending upon conditions of the contract. In addition, the PO monitors the contractor at this time through PQTs identified and scheduled in the development contract. PQTs initiate the formal test phase of DT&E; formal in the sense of Air Force participation. The objective of the PQTs is to provide the PO with visibility during this period. A PQT is a performance-level test of the CPCI or of a functionally related group of CPCs. For example, PQT is conducted on a function or group of functions prior to Formal Qualification Test (FQT), for instance, on the Weapons Guidance Function. These tests are conducted against the CPCI Development Specification requirements and are used to instill confidence that the CPCI will satisfy its performance requirements at the time of formal qualification.

2.4.1.3.2 QA Considerations

The concept of PQTs can be a very effective monitoring tool if implemented properly; if not, it can increase costs and decrease the SD's confidence in the development process. The number of PQTs conducted should be based on:

- Complexity of the CPCI
- PO manpower capabilities and availability
- Level of confidence in the contractor

The contractor's test documentation should be reviewed for:

- Ability of portions of the CPCI to achieve their performance objectives.
- Adequacy of the test methods and procedures to produce results required by the individual PQTs.
- Conformity of test results with performance requirements (Development Specification).
- Compliance with prescribed test procedures .
- Assurance that all corrective action has been taken by the contractor.

2.4.1.4 CPCI Qualification and Acceptance

The purpose of CPCI qualification and acceptance (see Task 14, Acceptance, ESDM 74-1) is to demonstrate that the CPCI performs in accordance with its CPCI Development (Part I) Specification and that all contractual requirements have been satisfied prior to acceptance of the CPCI and its related documents by the Air Force. The associated blocks in the Full-Scale Development Phase flow chart (see Figure 4) are:

- Prepare FQT Procedures - BLOCK D-16
- Accomplish CPCI Adaptation, Installation, and Check-Out* - BLOCK D-17
- Complete CPCI Product (Part II) Specification(s) - BLOCK D-18
- Conduct FQTs - BLOCK D-19
- Prepare CPCI Qualification Final Report - BLOCK D-20
- Issue Handbooks and Manuals - BLOCK D-21
- Conduct Functional Configuration Audit (FCA) - BLOCK D-22
- Conduct Physical Configuration Audit (PCA) - BLOCK D-23

*Adaptation refers to the parameters used to tailor the CPCI to the unique requirements of a particular location. Installation and check-out refers to the activities associated with the installation and testing of the CPCI at a particular site.

2.4.1.4.1 Summary of Activities

Prior to the PO accepting the contractor-developed CPCI and related data items, the contractor must first:

- Demonstrate that the CPCI performs in accordance with the approved CPCI Development Specification.
- Prove that the technical documentation is accurate and compatible with the qualified CPCI.
- Verify the accuracy of the configuration management records.

PQTs are normally conducted at the contractor's development facility. However, FQTs are normally best conducted at the System DT&E site (Category II test site). The contractor will adapt his CPCI to the site environment and install and check it out prior to initiating FQT.

FQT is a comprehensive performance test of the integrated CPCI to verify compliance with requirements of the CPCI Development (Part I) Specification. Once the CPCI has been qualified the CPCI Product (Part II) Specification is completed and support documentation updated to reflect the contents of the qualified CPCI.

2.4.1.4.2 QA Activities

QA activities for CPCI qualification and acceptance are performed in conjunction with FCA and PCA. The objective of FCA is to audit the results of the qualification tests to determine if the CPCI performance requirements have been met. In performing his QA responsibilities for FCA and PCA, the SD should determine if:

- The test reports are correct and valid.
- All known failures and nonconformances are reported, analyzed, and corrective action taken.
- All phases of CPCI testing have been completed.
- Test results conform with CPCI Development Specificaton requirements.
- Test procedures were properly checked before each test.
- Any deviations were requested and approved.

- The CPCI Product Specification is up-to-date and reflects all approved ECPs to the Development Specification.
- Configuration management records are accurate and up-to-date.
- All CDRL requirements have been satisfied.
- There is a list delineating all outstanding deviations against the CI, either requested or approved.
- The user/operator manuals and positional handbooks have been verified.
- All the delivered products are properly identified.
- The source code complies with standards.

2.4.2 Common Full-Scale Development Phase QA Problems and Proposed Solutions

Some of the common problems experienced on CPCI development contracts are:

- Establishment Of The Allocated Baseline Delayed Until PDR. Requires the contractor to provide a CPCI design approach based on a set of unapproved requirements. The requirements are often changed during approval which in turn may cause a major redesign and a slip in schedule.
- Too Many PQTs. The packages identified for PQTs are often small and many. Having scheduled many PQTs, the CDRL normally calls for the associated formal documentation, i.e., test procedures and reports, thus increasing data costs. The intent of the PQTs is to allow the SD to monitor the performance of the CPCI and build confidence in it as it is being developed. One PQT every few months may be sufficient.
- Wrong Types of Listings and Inadequate Level of Detail in Product Specification Flow Charts. Within the CPCI Product Specifications two areas in particular may cause problems at the time of delivery: (1) level of detail within the flow charts and (2) types of listings delivered. These two areas have created a considerable amount of unnecessary cost and misunderstanding between the PO and contractors. For example, the PO expects one level of flow charts and the contractor has planned and costed a different level.

- PO Attempts to Control Contractors. POs attempt to get control over contractors by planing interim baselines between the Allocated Baseline and product acceptance. This practice can create contractual problems and may legally relieve the contractor of his responsibility for meeting the Development Specification. This problem can be caused by giving design approvals at PDR or CDR. In some cases it has been caused by not sequencing the design reviews and audits properly. Design reviews are intended to monitor the contractor's technical progress. The minutes are the only item approved. This approval means that the minutes accurately reflect what happened at the review. (See Reviews and Audits guidebook and Monitoring and Reporting Software Development Status guidebook for detailed discussions of this subject.)

- ECPs Impacting Costs and Schedules. Some contractors bid low and increase cost through changes. ECPs should not be evaluated strictly on technical merits. The PO must place heavy emphasis on the cost and schedule impacts of each change. The PO should always ask the question, "What does it cost if the change is not made?"

Many problems experienced during Full-Scale Development can normally be traced to the following sources:

- Inadequate Development Specification.
- The use of generalities when specifying deliverables on contract.
- Not selectively applying the requirements of the military RSSs to the needs of the program/contract.
- Lack of fully understanding the contractor's/PO's rights and responsibilities on the Development Phase contract.
- Incompatibility of hardware and software development schedules.
- Insufficient attention to the software development approach.

- Contractor is Required to Conduct Development Effort at a Location Remote from Home Office. The contractor often has difficulty relocating key personnel.

2.4.2.1 Proposed Solutions

Applying the following guidelines will assist the SD in minimizing Full-Scale Development Phase problems:

- Always establish the Allocated Baseline prior to PDR.
- Check with the contractor to ensure that CPT&E has been accomplished according to the CPDP.
- Do not schedule too many PQTs.
- If there is any difficulty regarding the cost of data and all the documentation normally associated with PQTs, schedule the tests and do not call for all the formal documentation on the CDRL.
- Always insist that a CPCI Product Specification data item description has a back-up sheet. The back-up sheet should be approved during Full-Scale Development Phase contract negotiations and should clearly define (1) level of detail of the flow charts and (2) types of listings to be delivered.
- To solve the problem of increased costs through ECPs, don't authenticate CPCI Development Specifications that do not precisely state requirements. Generalities and ambiguities within specifications encourage interpretation problems that require changes and clarifications. When reviewing proposals, beware of an underestimated effort.
- If possible, allow the contractor to propose the location of his Full-Scale Development Phase personnel. As remote computing capabilities become widespread it is no longer necessary to locate development programmers at the site of the development computer.

SECTION 3 - CONTRACTOR SOFTWARE QUALITY ASSURANCE PROGRAMS

3.1 INTRODUCTION

This section is designed to assist the SD in evaluating a contractor's proposal and monitoring the status of a contractor's software QA program.

The basic military specification concerning contractor software QA programs is MIL-S-52779(AD)*. This specification takes the hardware-oriented requirements of MIL-Q-9858A and adapts them to software with the overall objective of providing a software acquisition management tool that can be selectively applied to software development programs by the PO. QA requirements, therefore, must be tailored to the needs of each program. The intent of this section is to identify a series of questions which should be askeu by the SD when evaluating the contractor's software QA program.

3.2 SOFTWARE QUALITY ASSURANCE RESPONSIBILITIES

The QA responsibilities of all organizations participating in the software acquisition process must be clearly understood. The PO must first establish the requirements for the QA program and specify the requirements in the contract. Then the PO must determine if the contractual requirements have been met prior to acceptance. The contractor, therefore, is responsible for implementing QA procedures which will assure that the requirements of the contract are satisfied.

3.3 PRE-CONTRACT AWARD

The success or failure of the Full-Scale Development Phase contract is usually determined by the quality of results obtained from the previous phases and by how well the results have been reflected as requirements within the Full-Scale Development contract. The key items of concern in the basic Full-Scale Development contract are:

- The Statement of Work.
- The CPCI Development (Part I) Specification(s).
- Selection and tailoring of appropriate specifications and standards.
- Selection and tailoring of the Data Item Descriptions (DIDs) listed in the CDRL.
- The compatibility of CPCI(s) development with the total system development, as reflected in the contract plans. Such plans include the CPDP and the CPCI Qualification Test Plan [Category I Test Plan/Procedures (Computer Program)].

*MIL-S-52779(AD) may be in conflict with AFR 74-18 which controls (hardware) QA during acquisition. This guidebook is written in accordance with MIL-S-52779(AD).

Attention should be focused on the guidebooks which provide information on the Full-Scale Development Phase contract and its related documents. These guidebooks include:

- Contracting for Software Acquisition
- Statement of Work Preparation
- Requirements Specification
- Software Documentation Requirements

3.4 IMPLEMENTING THE CONTRACTOR'S QA PROGRAM

The contractor's QA program consists of a combination of internal management practices, procedures, and controls which are the techniques he uses to direct and control development efforts. Performance requirements and design constraints, such as growth potential to facilitate modification and expansion, use of structured programming techniques, build approaches to integration and test, are not part of the contractor's QA program and should not be confused as such (software QA vs quality software).

In evaluating the contractor's QA program against the contractual requirements of MIL-S-52779(AD), the following basic questions should be asked by the SD:

- Does the contractor have a software QA program which assures compliance with the requirements of the contract?
- Is the program documented and is such documentation available to the Government?

Further comprehensive guidance on this subject can be found in SAMSO Pamphlet 74-2.

In addition, although there is no standard DID for acquiring a software QA program, the Software Documentation Requirements guidebook provides instructions for modifying the CPDP DID to include a software QA plan.

The remainder of this subsection provides a series of checklists that the SD can use in evaluating the contractor's software QA program in the following areas:

- QA organization and authority
- Work tasking and authorization procedures
- Configuration management
- Testing

- Corrective action procedures
- Library controls
- Computer program design
- Software documentation
- Reviews and audits
- Tools, techniques, and methodologies
- Subcontractor control

3.4.1 QA Organization and Authority

The implementation of an effective software QA program requires contributions from all contractor organizations associated with the project. Contractors are expected to establish a QA organization which is responsible for overseeing the implementation of the QA program. In reviewing this element of the overall QA program the SD must determine the following:

- Has the contractor identified the organizational elements responsible for software QA?
- Do the personnel performing the software quality functions have sufficient authority, responsibility, and freedom of action to evaluate software design and development activities, and to initiate and/or recommend changes?
- Is the staffing adequate and are the personnel qualified to perform the QA role?

3.4.2 Work Tasking and Authorization Procedures

The contractor's QA procedures for issuing work tasking instructions should provide for definition and authorization of tasks, tracking and reporting task progress, resource allocation, and steps for closing out completed tasks. The following questions should be answered during the evaluation of these procedures:

- At what level within the organization are the tasks authorized?
- Is the level of authorization sufficient to provide management control?
- Are there provisions for monitoring and tracking the progress of tasks?
- Can the task progress be related to the approved project schedules?
- Is the relationship between tasks and WBS elements visible?
- Do the tasking procedures call for a detailed description of the tasks related to the SOW?

- Is the responsible manager for each task identified?
- How is the allocation of resources accomplished?
- What procedures are followed regarding the close-out of completed tasks?

3.4.3 Configuration Management

In evaluating the contractor's proposed QA program, the SD should determine answers to the following questions:

- Does the contractor's configuration management plan satisfy the requirements of MIL-STD-483, Appendix I?
- Does the configuration management plan provide adequate internal controls to ensure that no unauthorized changes occur to baseline specifications, supporting documentation (e.g., test plans, user manuals), or the CPCI?
- Does the software QA program require audit of configuration management procedures and practices?
- Are the results of the audits documented and maintained for SD review?

3.4.4 Testing

The contractor's test planning information should not be included or duplicated in the software QA plan. The contractor's test plans and practices should be documented in the CPDP and in the CPCI (Category I) DT&E Plan. These documents should be reviewed when evaluating the QA aspects of the test program and answers to the following questions determined:

- Does the software QA program identify the contractor's software test activities?
- Has testing responsibility been identified and assigned to a specific organization?
- Does the contractor have procedures and documentation controlling his internal CPT&E activities?
- Have the various levels of tests been identified and scheduled?
- Have PQTs and FQTs been scheduled to provide visibility into the contractor's test effort?
- Does the software QA program provide for review of test plans for compliance with contractual requirements?

- Does the software QA program provide for review of test procedures for compliance with appropriate standards and satisfaction of contractual requirements?

- Does the software QA program provide for monitoring of tests and certification that test results are the actual findings of the tests?

- Is test-related documentation maintained to allow repeatability of tests?

- Is all support software and computer hardware used to develop the CPCI, qualified and accepted by the Government?

3.4.5 Corrective Action Procedures

The contractor is required to delineate procedures which will assure the prompt detection, communication, and correction of deficiencies and errors. These procedures are intended to avoid noncompliant CPCIs and should identify:

- Methods of reporting and analyzing problems.

- Methods of communicating the problems and their resolution.

- Techniques used for statusing problems and implementing solutions.

- Methods used for conducting trend analyses and reviews of the effectiveness of the corrective action program.

- Corrective action procedures to be imposed on subcontractors.

3.4.6 Library Controls

A critical part of the contractor's QA program is the set of procedures used to control the source code and object code in their various forms during development and test activities. In reviewing the contractor's QA program, the SD should determine answers to the following questions:

- Has the contractor established a computer program library to be used for controlling program materials during development and test? (One type of library is that envisaged by RADC-TR-74-300, Volume VI, which describes the program support library.)

- Do the procedures identify how materials are approved and placed under library control?

- Do the controls include formal release procedures for internally approved design information?

- What safeguards have been established to assure that no unauthorized alterations are made to the controlled materials?

- Do the procedures indicate that all approved modifications are integrated?

3.4.7 Computer Program Design

The software QA program requires that the contractor establish technical control and evaluation of his products as they are being developed. This requires the contractor to establish procedures for reviewing and evaluating the CPCI design and associated documentation as they are being developed and prior to release. These procedures should be reviewed by the SD to determine:

- Do the contractor's procedures address the conduct of internal design reviews?
- Are the formal and informal reviews scheduled at critical decision points during development?
- Are design problems identified and followed-up for complete corrective action prior to approval of design?
- Are design reviews conducted prior to release for coding?

3.4.8 Software Documentation

The contractor's QA program should identify standards and procedures which assure delivery of accurate and up-to-date documentation. The SD should review these QA procedures to determine:

- Does the contractor identify standards to be followed when preparing the required documentation?
- Do the procedures call for technical review of documentation prior to release?
- Do the procedures address the control of changes to software documentation?
- Are there provisions for informing design personnel of the latest changes to the software documentation?
- Do the procedures provide for traceability of changes?

3.4.9 Reviews and Audits

The contract normally calls for reviews and audits at fixed points during the software development process. Reviews are usually conducted by the contractor with the PO in attendance. Audits are conducted by the Government with possible contractor assistance (see Reviews and Audits guidebook). The requirements for the conduct of reviews and audits are specified in MIL-STD-1521A (USAF). The contractor's CPDP should reflect the scheduling to review their preparation and conduct. The SD should review these documents to determine:

- Are the reviews and audits clearly identified, scheduled, and properly sequenced?
- Will the reviews and audits be conducted within the guidelines of MIL-STD-1521A(USAF)?
- Do the procedures define the types of information to be presented at each review?
- Are there agreements for follow-up action resulting from the reviews and audits?
- Will the results of the reviews and audits be documented by the contractor?

3.4.10 Tools, Techniques, and Methodologies

The software QA organization should review the techniques and tools the contractor plans to use in support of his developmental activities. These techniques, methodologies, and tools range from systems and engineering methodology (including software engineering techniques and methods), to support tools for developing and testing software. In reviewing the contractor's QA program, the SD should determine answers to the following questions:

- Has the contractor identified and defined the system/software engineering techniques and methodologies he plans to employ? (They should be documented in his CPDP.)
- Are the contractor's proposed automated tools qualified or will they be qualified prior to use?
- Are the proposed automated tools documented and placed under configuration management control?
- Is the contractor experienced in the use of the proposed tools, techniques, and methodologies?
- Will the Government have access to all development tools required during deployment?

3.4.11 Subcontractor Control

It is the responsibility of the prime contractor to assure that his subcontractors conform to contract requirements. The software QA program should identify techniques employed by the prime contractor for controlling and monitoring all subcontractor activities. The SD should review these procedures to determine their adequacy. This review should determine the following:

- Does the prime contractor require subcontractors to prepare and maintain (1) a QA plan, (2) a CPDP, and (3) a configuration management, plan?
- Does the prime contractor review and approve his subcontractors' plans?
- Is subcontractor documentation submitted to the prime contractor for review and approval prior to release to the Government?
- Does the prime contractor participate in the subcontractors' design reviews and audits?
- Does the prime contractor monitor software subcontractor testing activities?

SECTION 4 - SOFTWARE QUALITY ASSURANCE AT ESD

4.1 INTRODUCTION

This section describes how ESD assists its POs in meeting their QA requirements. Within ESD the Directorate of Computer Systems Engineering (MCI) is responsible for assigning computer system personnel to each of the POs. Through a series of management initiatives which have evolved and improved during the past several years MCI is developing the checks and balances needed for software QA. The following features of MCI's approach contribute to software QA at ESD:

- The matrix management organization
- Computer Systems Evaluation Panel (CSEP)
- The Critical Assessment Factors (CAF) System and the Computer Resources Management Center (CRMC)
- Documentation Review
- Lessons Learned

4.1.1 Evolving QA Role

The above features are undergoing constant review and revision by ESD. They were initiated to respond to known requirements for improving the software acquisition process. They are being revisec to address changing requirements and to better improve the process. For example, the CAF System is now being expanded with check lists for each CAF and with references to portions of SAM Guidebooks which address each CAF. There is a question whether the MCI role will be a temporary or a continuing one but in either case the requirements will continue for an ever improving software QA function.

4.2 THE MATRIX MANAGEMENT ORGANIZATION

MCI is responsible for all ESD personnel who have computer systems job classifications. More than half of those personnel work directly for MCI and all of them provide support to the POs. MCI employs a matrix management organization concept whereby at least one individual from MCI (designated the Key Person) is assigned to each PO. His primary assignment is as a working member of the PO. In some cases he may be the SD. He is a member of the MCI organization and has an independent reporting function to MCI. This concept offers the following advantages:

- A centralized manning and career monitoring organization for ESD computer systems personnel.

- Centralized training to assure a common basic background for all ESD computer systems personnel.

- A planned rotation of personnel from MCI to direct assignments in the POs.

- An organized approach to a consistent software acquisition management philosophy, with attention given to lessons learned from past procurements.

Personnel assigned from MCI to the POs are responsible for reporting software status information both to the PO director and to MCI management. Monitoring attention is focused upon 36 Critical Assessment Factors (CAFs) which are designated for management attention because of their criticality to software acquisition (see Figure 5).

The matrix management concept used by MCI encompasses many attributes which contribute to software QA. The most important of these attributes is the independent reporting required of the MCI Key Person to both the PO director and to MCI management. The Key Person can succeed in his assignment only if he is able to identify potential problem and risk areas early enough for appropriate action to be taken. Another attribute is the improved ability of management to assign the appropriate person because management is responsible for a large pool of professionals.

4.3 COMPUTER SYSTEMS EVALUATION PANEL

By order of the Commanding General of ESD, the CSEP has been established to review each acquisition involving computer resources, at least twice, prior to contract award. The objectives of these reviews are to assure that the computer/software portion of the RFP is stated in a feasible and realizable management structure and project structure, and that cost, schedule, and performance parameters are realistic; and to also assess the effectiveness of the offeror's response to the RFP in regard to these same factors.

The first review takes place prior to the release of the RFP. It examines the computer/software aspects of the RFP package and the acquisition strategy as they relate to schedule, cost, and technical feasibility.

The second review takes place during the source selection process. During this review, the CSEP serves as an ad hoc group to the Source Selection Advisory Council (SSAC). The Panel is responsible for monitoring and overseeing the Source Selection Evaluation Board's (SSEB's) evaluation of the computer/software issues within the proposals under evaluation.

The CSEP uses a source selection checklist to assist in evaluating RFPs and proposals. The checklist highlights areas which were not as thoroughly reviewed in past procurements. These areas include:

- Schedule risks.
- Technical risks.
- Adequacy of the stated requirements.
- Conformance with regulations, specifications, and standards.
- Computer program products that will be useful to the Air Force throughout the system life cycle.

The checklists are comprehensive and include considerations which may receive various answers depending upon the characteristics and requirements of each procurement. To successfully achieve its objectives the CSEP must have access to individuals who have mature judgement in both the procurement and the technical aspects of software acquisition management.

4.4 THE CRITICAL ASSESSMENT FACTORS SYSTEM AND THE COMPUTER RESOURCES MANAGEMENT CENTER

The CAFs are 36 defined events or products that are essential to the system acquisition process and indicate the current status of the software in the PO (see Figure 5). The primary purpose of the CAFs is to advise the program manager on the status of computer hardware and software. The CRMC is a location where the collection, storage, and retrieval of reports, schedules, and procedures to support the CAF System is accomplished.

The MCI Key Person assigned to each PO is responsible for preparing CAF reports and for providing supporting information for use in the CRMC. Each CAF receives one or more ratings from the Key Person. The ratings, which are based primarily on consideration of impact upon schedule, cost, and performance, indicate a status of satisfactory (green), marginal (yellow), or unsatisfactory (red). His responsibilities include:

- Submitting a CAF report schedule.

- Submitting the CAF reports

- Ensuring that the CAF reports are properly routed.

- Responding to all CAF delinquency reports.

The matrix assignment of the Key Person from MCI provides an independent review from an individual whose assignment requires him to be intimately involved with the details of the procurement.

There are two types of CAF reports, Status and Early Warning. The Status Reports discuss present or past events and the Early Warning Reports make predictions about future events. Both types of CAF reports contribute significantly to software QA at ESD. They focus attention on the most important products and events and provide for a regularly scheduled management review at a high level of visibility.

Further information is provided by trend arrows which indicate the expected stability of the CAF rating and predict the next CAF color. The CAF System provides several levels of timely management attention. The initial report goes to both the Program Director (PD) and MCI. Reviews are made by both the PD and the MCI management. Reports are made at least monthly. The CAF system provides:

- Early and regular visibility.

- Pressure to resolve problems at lowest levels.

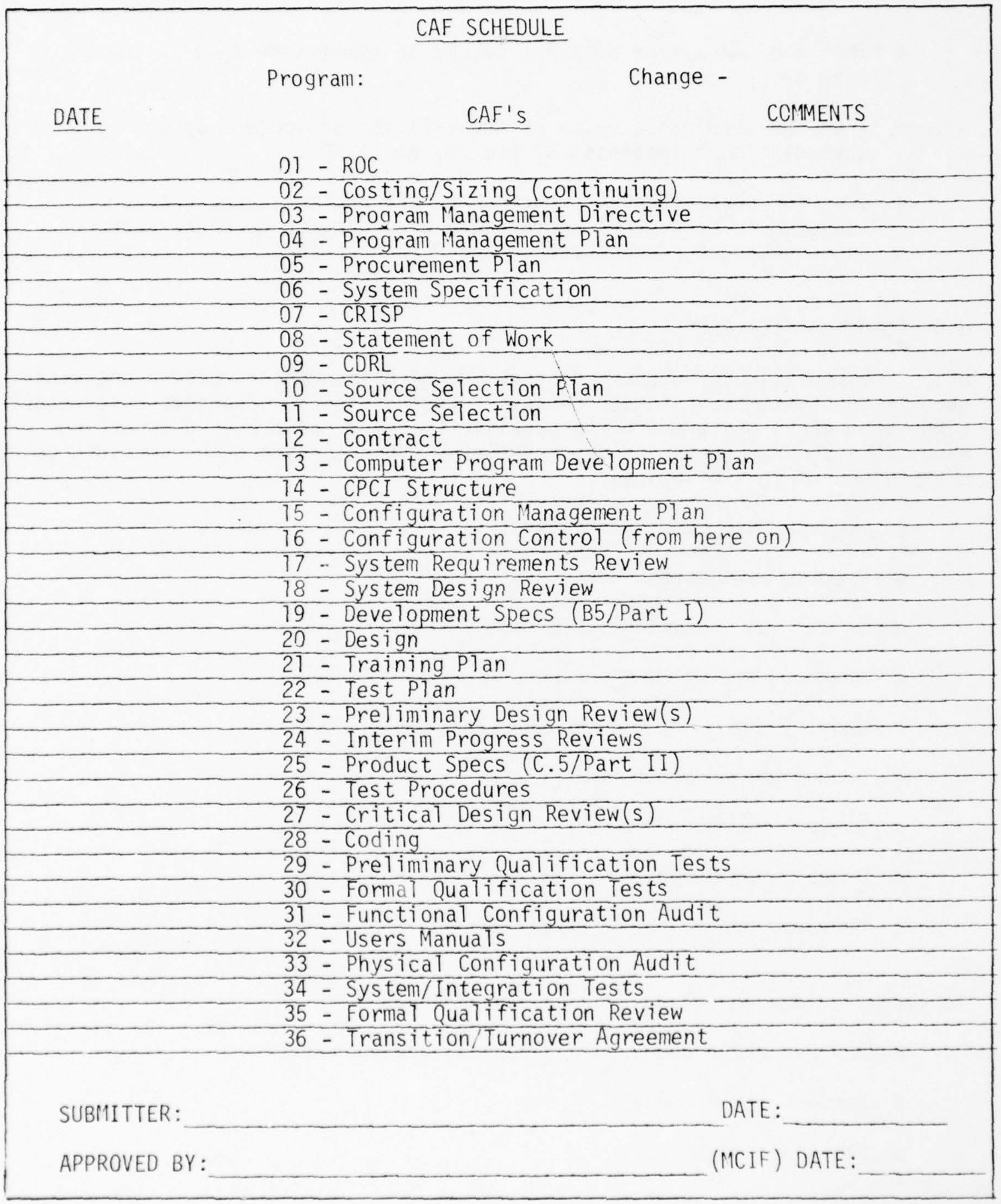
- Machinery to escalate to highest levels as appropriate.

Figure 5 shows the standard CAF schedule which may be modified to meet specific program needs. It lists the 36 CAFs as they currently exist. Some of the CAFs (such as 02-Costing/Sizing) are continuous throughout the program. Most of the CAFs are reported upon only at the time of their expected and actual occurrence.

To retain its full impact as a QA tool the CAF System should be under constant MCI management review to determine:

- That the CAFs are consistently understood and used by the Key Persons, by MCI, and by the PDs.

- That the CAFs receive an ongoing review and definition revision to ensure that the apparent simplification of the system does not obscure the complex issues it is intended to rate.



CAF SCHEDULE

Program: Change -

DATE	CAF's	COMMENTS
	01 - ROC	
	02 - Costing/Sizing (continuing)	
	03 - Program Management Directive	
	04 - Program Management Plan	
	05 - Procurement Plan	
	06 - System Specification	
	07 - CRISP	
	08 - Statement of Work	
	09 - CDRL	
	10 - Source Selection Plan	
	11 - Source Selection	
	12 - Contract	
	13 - Computer Program Development Plan	
	14 - CPCI Structure	
	15 - Configuration Management Plan	
	16 - Configuration Control (from here on)	
	17 - System Requirements Review	
	18 - System Design Review	
	19 - Development Specs (B5/Part I)	
	20 - Design	
	21 - Training Plan	
	22 - Test Plan	
	23 - Preliminary Design Review(s)	
	24 - Interim Progress Reviews	
	25 - Product Specs (C.5/Part II)	
	26 - Test Procedures	
	27 - Critical Design Review(s)	
	28 - Coding	
	29 - Preliminary Qualification Tests	
	30 - Formal Qualification Tests	
	31 - Functional Configuration Audit	
	32 - Users Manuals	
	33 - Physical Configuration Audit	
	34 - System/Integration Tests	
	35 - Formal Qualification Review	
	36 - Transition/Turnover Agreement	

SUBMITTER:____________________ DATE:________

APPROVED BY:____________________ (MCIF) DATE:________

Figure 5. CAF Schedule

- That the CAF System does not become so cumbersome that it is not used or updated.

- That the predictive value of the CAFs is not impeded by the jurisdictional interests of the POs or of MCI.

- That for each specific program, the CAF System's coverage is tailored to review all appropriate areas (e.g., when software is developed by two contractors).

4.5 DOCUMENTATION REVIEW

MCI is responsible for performing a review of all computer-related documents used in the system acquisition process. Each review is assigned to personnel other than those assigned to the specific PO. MCI has produced a working paper (MCI-75-002) to assist the reviewers. This working paper provides brief discussions of the following:

- Program Management Directive
- Program Management Plan
- Individual Contract Procurement Plan
- Advanced Procurement Plan
- Justification for Authority to Negotiate
- Determination and Finding
- Source Selection Plan
- Request for Proposal
- Required Operational Capability

It also provides checklists for review of the:

- RFP Executive Summary and Proposal Instructions
- Request for Proposal
- Statement of Work
- Contract Data Requirements List
- System Specification

4.6 LESSONS LEARNED

MCI has published an internal document entitled "Lessons Learned." It distills experiences from past procurements and identifies specific problem areas which have occurred. It provides suggestions on how to avoid similar problems in future procurements.

While "Lessons Learned" can only be loosely described as a QA tool, it does assist in providing a "Corporate Memory" for ESD upon which the QA reviewer can isolate potential problem areas.

APPENDIX A - SOFTWARE QUALITY ISSUES

This appendix identifies and discusses the following major software quality issues:

- Software quality
- How much QA is enough?
- The case for independent technical support contractors.

1. SOFTWARE QUALITY

Software quality is a composite of many factors, some of which conflict with each other (e.g., efficiency and maintainability). To further complicate the issue, many attributes of software quality can only be assessed after delivery and during use.

To assure the development of quality software, the SD needs to establish confidence in quality through the basic development process, knowing from past experience that if the basic functions and policies of the acquisition process are properly used, software quality will be predictably good. It is, therefore, the sole purpose of the QA program to ensure that this happens. To accomplish this the PO must monitor, review, and evaluate in depth every aspect of the computer program life cycle. Subsequent discussion in this appendix, therefore, focuses on the following software quality-related issues:

- What is software quality?
- Is software quality measurable?
- Should questionable quality cause program delays?

1.1 WHAT IS SOFTWARE QUALITY?

The question of what constitutes software quality is currently undergoing vigorous discussion within both industry and military R&D organizations. Many software quality characteristics are specified by the military in terms of performance requirements. Others do not fit into system acquisition terminology. Currently, many of these characteristics, as such, do not apply to system acquisition because they are reflected in the software design and supporting documentation rather than in the performance requirements. This discussion identifies some of these software quality characteristics.

Figure 6 shows software quality as a three-level hierarchy with user-oriented quality attributes in Levels 1 and 2 and examples of the software characteristics required to provide these attributes in Level 3.

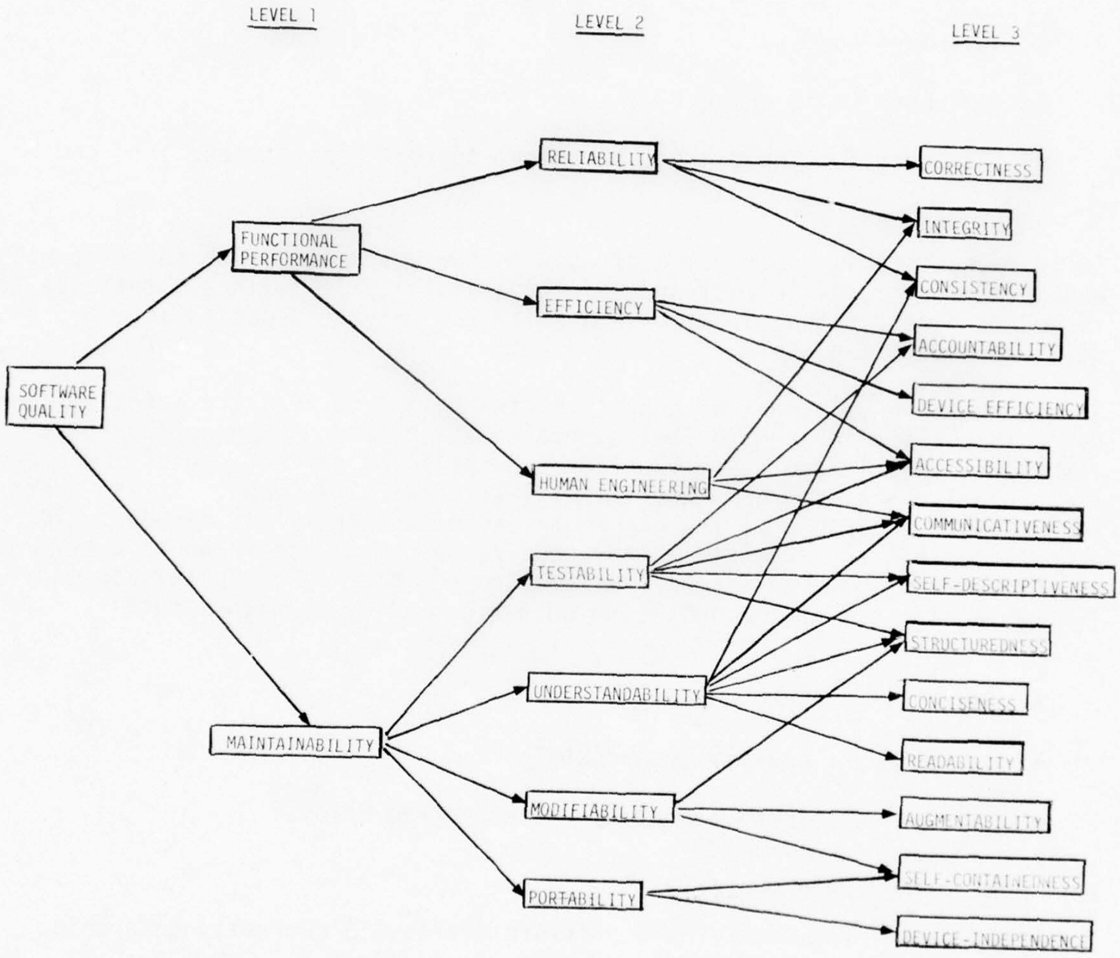


Figure 6. Characteristics of Software Quality

Definitions of each of the terms shown in Figure 6* can be found in Appendix B. Many of these definitions overlap. In addition, some (e.g., reliability and maintainability) are inconsistent with military use of these terms. They are included in this guidebook to reflect common software industry usage.

1.2 IS SOFTWARE QUALITY MEASURABLE?

The evaluation of software quality is based upon testing (which is almost always non-exhaustive) and upon more subjective evaluations of software quality, such as inspection, for maintainability and other attributes. However, many attributes of software quality can only be assessed after delivery and during use. Thus, software quality can only be partially measured. Figure 7 illustrates and prioritizes the methods used for measurement and evaluation of quality for selected development activities of the acquisition cycle.

	DEVELOPMENT ACTIVITIES					
EVALUATION METHODS	VERIFY PERFORMANCE REQUIREMENTS	AUTHENTICATE DEVELOPMENT SPECIFICATION	PDR/CDR	PQT/FQT	PCA	SYSTEM DT&E
INSPECTION				3	1	3
ANALYSIS	1	1	1	2	2	2
SIMULATION	2			4		4
TESTING				1		1
REVIEW	3	2	2			

Figure 7. Priorities in Evaluation of Quality.

*Slightly modified figure from "Quantitative Evaluation of Software Quality." (See Appendix C.)

Both the Government and industry have performed several research and development studies aimed at defining metrics for the measurement of software quality. The following is quoted from a recent report developed for the National Bureau of Standards:

> "Calculating and understanding the value of a single, overall metric for software quality may be more trouble than it is worth. The major problem is that many of the individual characteristics of quality are in conflict: added efficiency is often purchased at the price of portability, accuracy, understandability, and maintainability; added accuracy often conflicts with portability via dependence on word size; conciseness can conflict with legibility. Users generally find it difficult to quantify their preferences in such conflict situations. Another problem is that the metrics are generally incomplete measures of their associated characteristics. To summarize these considerations:
>
> - The quality of a software product varies with the needs and priorities of the prospective user.
> - There is, therefore, no single metric which can give a universally useful rating of software quality.
> - At best, a prospective user could "...achieve a meaningful..." rating system with a thorough set of checklists and priorities.
> - Even so, since the metrics are not exhaustive, the resulting overall rating would be more suggestive than conclusive or prescriptive.
> - Therefore, the best use for metrics at this point is as individual anomaly indicators, to be used as guides to software development..."

In a more recent paper given at the 2nd International Conference on Software Engineering, these same authors more optimistically concluded:

- "Explicit attention to characteristics of software quality can lead to significant savings in software life-cycle costs.
- The current software state-of-the-art imposes specific limitations on our ability to automatically and quantitatively evaluate the quality of software.

- A definitive hierarchy of well-defined, well-differentiated characteristics of software quality has been developed. Its higher-level structure reflects the actual uses to which software quality evaluation would be put; its lower-level characteristics are closely correlated with actual software metric evaluations which can be performed.

- A large number of software quality-evaluation metrics have been defined, classified, and evaluated with respect to their potential benefits, quantifiability, and ease of automation.

- Particular software life-cycle activities have been identified which have significant leverage on software quality. These include:

  - Setting explicit software quality objectives and priorities;
  - Performing software quality benchmarking;
  - Using software quality checklists;
  - Establishing an explicit quality assurance activity;
  - Using machine-analyzable software specifications;
  - Ensuring testable software requirements;
  - Using a Requirement-Properties Matrix;
  - Establishing standards, particularly for structured code;
  - Using an automated Code Auditor for standards compliance checking;
  - Performing design and code inspections."

Other industry research* reflects similar conclusions regarding the measurement of software quality. In summary, these conclusions are:

- The measurement of software quality is still in the R&D stage.

- Software quality is measurea in terms of quality characteristics (such as those shown in Figure 6). Those characteristics need to be more clearly defined in a more mutually exclusive and meaningful manner.

- The relative priority of characteristics varies between applications and between users.

- Measurement of software quality is a promising area for better definition and acquisition of quality software.

*From "Quantitative Measurement of Program Quality" and "Factors in Software Quality." (See Appendix C.)

1.3 SHOULD QUESTIONABLE QUALITY CAUSE PROGRAM DELAYS?

A primary role of the QA program is to identify problems before they become serious. The SD's most complex job, from a technical as well as a managerial viewpoint, is to prevent software related problems from getting out of control.

Many problems can only be resolved by schedule delays (e.g., disapproval of specifications, disapproval of documentation until all undefined areas are removed, disapproval of FQTs). However, most delays cost money and have potential impact on overall system development activities. Prior to any such recommendation, the SD should identify the problem, investigate alternatives with the contractor, select an approach for resolving the problem, agree upon schedules for resolution of the problem, and carefully monitor implementation of the resolution.

Early computer program life cycle activities (e.g., requirements specification, CPCI functional definition) have a major impact on software quality. However, there is a natural inclination to resist actions which can cause schedule slips during these early phases since the apparent impact of the problem is not always obvious. Early delays may have financial and schedule impact although generally small in comparison to the results of ignoring early indications of trouble. In fact, if a delay does not impact the critical path of the overall system development, it may save money because it prevents additional work from starting before a solid base is constructed.

A well planned QA program emphasizes the use of techniques which simplify review and provide early visibility into the development process; once again, quality must be built into the software. Techniques which provide this simplification and visibility include:

- Appropriate application of acquisition policy, stressing the use of software development phases and their associated reviews and audits.
- Emphasis on clear and complete System and Development (Part I) Specifications.
- Implementation of the software in increments or builds. (See Monitoring and Reporting Software Development Status guidebook.)
- Use of a top-down development philosophy. (See Monitoring and Reporting Software Development Status guidebook.)
- Use of simulation and prototype development for design verification.
- Use of program production libraries*.
- Selection of computer program components for PQT that will build confidence in the developing CPCI.

*Also called program support libraries.

Difficult delay decisions are often required at FQT and during System DT&E. When a CI is delayed and becomes unavailable for system integration, the entire System DT&E schedule may be affected. To avoid this problem the SD should identify those CPCIs which are most critical to system integration schedules and request their demonstration for PQTs. If possible, FQTs for those CPCIs should be scheduled first so that delays will not impact system integration schedules.

2. HOW MUCH QUALITY ASSURANCE IS ENOUGH?

The nature of the software QA job is such that no matter how much effort is budgeted or expended, additional effort can always be applied. The software QA effort is applied by the PO, by the software developer, and in some cases by an independent support contractor [e.g., an Independent Verification and Validation (IV&V) contractor]. Their combined efforts make up the overall QA job. The SD must recommend how much software QA the PO can, or should, pay for, and then must allocate and contractually assign the QA functions to the organizations he believes are best able to accomplish them.

There are certain criteria that can be used in determining where increased or decreased software QA effort is merited. One is to perform a risk analysis of the impact of the software on the overall system. Whenever risk is great an intensive software QA effort is merited. Another criterion concerns analysis of the types and thoroughness of software testing available. For many systems it is difficult to design a thorough and realistic test program which provides sufficient confidence that the software will properly perform within its system operational environment. This is particularly true when multiple capabilities are to be tested or when complicated interactive environments (e.g., an ECM/ECCM environment) are required. In such cases, a thorough test program augmented by analytical and/or simulation methods is most desirable*.

Other risk factors which serve as criteria for increased emphasis on the QA program include:

- Complexity of software applications (e.g., real time constraints, complex algorithms, multiple processes).
- Amount of software; potential for error increases greatly with size.
- Stability of requirements.
- Uniqueness of application, i.e., Has this ever been done before?

*See the Verification guidebook for a discussion of the methods available.

- Lack of experienced personnel.
- Rushed development schedules.
- Mission criticality of the software.
- Lack of interface confidence, i.e., are the interfaces with other CIs incompletely defined and/or likely to change?
- Immaturity or unavailability of computer hardware and support software.
- Suitability of the computer and programming language to the application.
- Unavailability of a realistic test environment.

If the SD must limit or focus the QA effort, he should emphasize quality of the Development (Part I) Specifications and of the interface definitions between CIs.

Another area of focus for QA is the review, test, and audit milestones of the System Acquisition Cycle.

The SD should critically evaluate the completeness and adequacy of:

- The Conceptual and Validation Phase trade-studies. (Have the risk areas been identified and limited?)
- The extent of prototype development or of performance simulation during the Validation Phase. (Are all the requirements deliverable within the state-of-the-art?)
- The use of simulation and modeling in design verification.
- The extent of the testing program in terms of the number of tests planned and the use of environmental simulators.
- The use of independent contractors to support the QA activity. e.g., an independent support contractor.

3. INDEPENDENT SUPPORT CONTRACTORS

For many years Air Force POs have used independent technical support contractors to assist in the planning and evaluation of software acquisition programs. The System Engineering/Technical Direction (SE/TD) contractor role (such as that performed by the MITRE Corporation) is well established and used on most major Air Force software acquisition procurements. More recently the IV&V contractor role has been defined and used in various ways. This discussion describes both the SE/TD and IV&V roles.

An analysis of Air Force system acquisition management effectiveness and the economy of using independent support contractors inevitably revolves around the question, "Is such support necessary, and if so, to what extent?" This question must be answered in terms of such program characteristics as system reliability requirements, software complexity, life cycle cost considerations, and the availability of in-house resources to monitor the acquisition.

The PO requires continuity of competent technical assistance throughout the System Acquisition Life Cycle. Technical monitoring is required during all portions of the software development process with emphasis on all documentation and test reviews. The PO has personnel available for technical monitoring; however, the experience level and continuity of the available technical personnel is not always sufficient. When additional technical support is needed, the scope of that support should be defined and can be procured from an independent contractor. The primary focus of the support effort should be upon obtaining the system engineering support needed to ensure complete and consistent requirements and for ensuring the testability of the Development (Part I) Specifications. Independent test should be a secondary consideration.

3.1 THE SE/TD ROLE

The SE/TD role has traditionally been focused at the system level rather than at the software level. The mission of the SE/TD is to provide scientific, engineering, and support personnel and facilities as an adjunct to Air Force resources. The SE/TD provides direct support to the program director in the acquisition of a system and in making studies and analyses of proposed systems. Their activities are usually concerned with developing the System Specification and then monitoring development activities to ensure system integrity.

Since the SE/TD works closely with the PO and is directly and intimately involved with advanced planning information, it is preferable that the SE/TD be a non-profit organization, or at least be forbidden from competing in any other aspect of the system acquisition.

The SE/TD role is a necessary and desirable one. It provides technical depth and continuity which is unquestionably beneficial to the Government's position. The only danger areas are:

- Possible complacency and lack of initiative on the part of an organization that has a continuing sole source role.

- Possible contractual conflicts with the development contractor [e.g., rights to data and direction of contractor personnel (see 3.3 of this appendix)].

3.2 THE IV&V ROLE

The IV&V contractor also provides support services to the PO in the area of technical evaluation and monitoring of acquisition activities. However, IV&V support is usually limited to software areas. In many cases, IV&V support tends to overlap the SE/TD role. IV&V support should always be tailored to the requirements of the specific procurement and may include:

- Operational Analysts/System Engineers. During the Validation Phase, IV&V contractor support may be required in the areas of (1) understanding the using command's problem from an operational standpoint and (2) providing system engineering capability and experience for defining CPCI performance requirements.

- Software System Analysts. Early in the Full-Scale Development Phase, IV&V software system analyst support may be required to evaluate the adequacy of the development organization's translation of performance requirements into a design approach. Independent design evaluation tools including simulation techniques, may be appropriate at this stage of development.

- Software Engineers. Later in the Full-Scale Development Phase, software engineering support may be required to review the adequacy of the detailed design of computer programs prior to coding. They can subsequently assist in reviewing Subsystem DT&E plans, procedures, and results and the software aspects of the System DT&E plans, procedures, and results. They can also provide technical support for the configuration management audits.

SAMTEC/LOGICON Report No. DS-R74036 discusses Independent Test and Evaluation (IT&E). The following paragraphs taken from that report define IT&E:

"IT&E is defined to be all analytical evaluations and tests conducted by an agency independent of the development contractor(s) to provide increased confidence that the software meets the system requirements. Typically, analytical evaluation includes reviews and analysis of requirements, documents, algorithms, equations, and code; tests include the review and active testing of the developed software by the IT&E agency. The primary objective (and therefore purpose) of IT&E is to assure the contracting agency that the delivered software:

- Is developed in accordance with the requirements as defined (and approved by the contracting agency) in System and Development Specifications for the software.
- Satisfactorily performs in the operational environment the functions for which it was designed.
- Does not perform unintended functions in the operational environment.
- Does not overtly or covertly degrade or limit hardware system or subsystem performance.

Other purposes of IT&E are to:

- Minimize development delays and costs by detecting errors early in software development.
- Evaluate the software's logical, mathematical, and coding design to optimize software performance.
- Provide the contracting agency with adequate visibility to ascertain the status of the software development throughout the development cycle.

In addition to fulfilling the objectives and purposes above, IT&E provides an element of competition that enhances the timely development of the software.

The ways in which IT&E programs can be constructed by the Air Force Program Manager depend on the nature of the software developed, the required level of confidence, and the existence of additional independent efforts being conducted on behalf of the procuring agency."

The LOGICON/SAMTEC report further describes four levels of IT&E which are identified as follows:

- "In Level 1, the IT&E agency is the integrating contractor for the hardware/software computer system. To this end, it performs all tasks necessary to integrate all CPCIs and segments into an efficient, operating computer system.

- In Level 2, the IT&E agency does not integrate the computer system. However, it performs all tasks necessary to independently test and evaluate the computer system to ensure that it has been developed according to SAMTEC-approved requirements and specifications. In addition, the IT&E agency assures SAMTEC that the computer system performs satisfactorily in the operational environment.

- Level 3 IT&E is similar to Level 2 except that the IT&E agency's level of effort is reduced to include only the most critical parts of the computer system. The SAMTEC Program Manager and the IT&E agency together identify the critical areas on which IT&E is concentrated.

- In Level 4, the IT&E agency's role is largely that of monitoring the software development. The IT&E agency assists the SAMTEC Program Manager in reviewing and evaluating the performance of the development contractor. This level does not include code analysis and software testing and is the most passive with respect to the development of the computer system."

The LOGICON/SAMTEC report also discusses costing parameters for IT&E.

3.3 RELATIONSHIPS BETWEEN INDEPENDENT SUPPORT CONTRACTORS AND THE DEVELOPMENT CONTRACTOR

The software development contractor is responsible for developing a CPCI that satisfies the performance requirements in the CPCI Development Specification. The support contractor is responsible for evaluating, monitoring, and recommending (through the PO) changes to the development contractor's technical activities. The independent contractor performs these tasks on behalf of the PO.

To facilitate the use of independent support contractors, the so-called "enabling" clauses must be incorporated in the development organization's contract to permit the support contractor access to technical data and participation in reviews, testing, and audits. (See Contracting for Software Acquisition guidebook and Statement of Work Preparation guidebook for a discussion of contractual relationships.)

In using independent support contractors, the PO must be aware of the following potential problem areas:

- Caution must be exercised by the SD to insure that the support contractor does not direct the development contractors. Otherwise problems in the area of "constructive changes" to contractual requirements may occur.

- The support contractor must not be permitted access to price information. The development contractor may be a competitor of the support contractor.

- Some contracts may make provisions for an independent support contractor to perform parallel testing of the CPCI during development. This method of operation can create legal and contractual problems if not handled properly. The PO cannot take a CI away from the development contractor and give it to another contractor without first "accepting" the CI. After acceptance the CI is government property and the PO may provide it to another contractor as GFP.

APPENDIX B - GLOSSARY

This appendix includes (1) definitions of major terms used throughout this guidebook, (2) definitions of terms used in Appendix A to discuss the broader issues of quality software vs software quality assurance, and (3) a list of acronyms and abbreviations used herein.

MAJOR TERMS

Authenticate. The act of signifying (by the approval signature of a responsible person of the procuring activity) that the Government is in agreement with the requirements contained in the specification. Authentication by the procuring activity normally will be accomplished on that issue of the specification which is to be the contractual requirement for the baseline which that particular specification defines [MIL-STD-483 (USAF) paragraph 3.4.9].

Computer Data. Basic elements of information used by computer equipment in responding to a computer program.

Computer Program. A series of instructions or statements in a form acceptable to computer equipment, designed to cause the execution of an operation or series of operations. Computer programs include such items as operating systems, assemblers, compilers, interpreters, data management systems, utility programs, and maintenance/diagnostic programs. They also include application programs such as payroll, inventory control, operational flight, strategic, tactical, automatic test, crew simulator, and engineering analysis. Computer programs may be either machine-dependent or machine-independent, and may be general purpose in nature or be designed to satisfy the requirement of a specialized process of a particular user.

Module. Used in this document to describe the smallest computer program unit that can be compiled or assembled. A CPC has one or more modules.

Program Support Library. A group of manual or automated procedures and tools used to control and keep records of the developing software.

Software. A combination of associated computer programs and computer data required to enable the computer equipment to perform computational or control functions.

Traceability. Refers to the capability to follow specific mission requirements through the various levels of specification to the actual code; and the capabilities to associate each area of code with a specified requirement.

Validation. Validation as used in this guidebook series comprises those evaluation, integration, and test activities carried out at the system level to ensure that the finally developed system satisfies the using command's mission requirements set down as performance and design criteria in the system specification.

Verification. The iterative process of determining whether the product of each step of the Cc nputer Program Configuration Item (CPCI) development process fulfills all of the requirements levied by the previous step.

QUALITY SOFTWARE TERMS

Accessibility is a characteristic of code or data which facilitates selective (and at times limited) use of its parts. It is a design rather than a performance characteristic and should not be specified.

Accountability refers to the ability to measure the computer hardware and peripheral equipment usage of a module or program.

Augmentability is a feature of the design and code which allows it to be easily modified. Elements of augmentability include modularity, parameterized data, and a centrally defined and controlled data base. Augmentability is primarily a design characteristic and should be proposed by the contractor.

Communicativeness is a feature of the software's inputs and outputs which facilitates understandability. Communicativeness enhances understandability and testability of the software.

Conciseness is the absence of redundant or excessive code.

Consistency of code means uniform standards for notation, symbology, terminology and comments. Consistency also measures the extent that the different representations of the software (System Specification, Development Specification, and Product Specification) are traceable to the requirements.

Correctness means the ability of the software to produce the specified outputs when given the specified inputs.

Device Efficiency is the optimized use of peripheral equipment and includes: Avoiding waste of peripheral storage space, performing data transfer quickly, and printing at optimum speeds. It can be specified in terms of specific measurable requirements.

Device Independence is the ability of the code to be unaffected by changes to the computer hardware or peripheral equipment. This quality indicates that code which is directly related to a specific hardware device should be minimized, isolated, and specifically identified. This quality can be specified as a design constraint.

Efficiency is the ability of the software to perform without waste of resources. Efficiency requirements can and should be specified in terms of performance requirements. It usually takes extra time and effort to design and develop efficient software. Efficiency is often sacrificed in order to enhance other software quality characteristics such as structuredness and readability.

Functional Performance is the ability of the software to satisfy its mission requirements as allocated from the System Specification and as contractually specified in the Development Specification.

Human Engineering is the design of interfaces between the software and the user. These interfaces should be specified in the Development (Part I) Specification. Inputs and outputs should be self explanatory, easy to learn and understand, unambiguous, and designed to avoid misinterpretation.

Integrity is a feature of the design and code that describes its uniformity of design and cohesiveness. Integrity is easiest to obtain when designing and developing software from scratch. It is more difficult to maintain integrity when off-the-shelf software is tailored to a new set of requirements or when a CPCI has undergone a series of changes. Software having integrity is less likely to contain errors and is easier to maintain.

Maintainability as applied to software is specification, design, and development of code in a manner which facilitates the task of modification to correct deficiencies and to satisfy new or changing requirements. A potential source of confusion exists regarding subtle distinctions between the hardware and software definition of maintainability. Hardware maintenance is the restoration of hardware to its original design, whereas software maintenance is defined as both error correction and modification of the original design (both of which imply change rather than restoration). Since there is little chance that the usage of either set of definitions will be discontinued, the SD should bear these differences in mind when participating in the establishment of maintainability criteria for the total system. Software maintenance features in terms of growth requirements may be specified in the Development (Part I) Specification. Additional features such as modularity should be requested in the RFP, responded to in the CPDP, and implemented by the contractor in the design, and reflected in the Product (Part II) Specification. (See Appendix A of the Software Maintenance guidebook.

Modifiability is a characteristic of the design and code that makes it easy to change. It is a difficult characteristic to specify and evaluate because objective measures of modifiability are not available. However, structured programming techniques include features (i.e., modularity, cohesiveness) which enhance modifiability. A qualified programmer can examine CPCs and judge their modifiability.

Portability is the ability to move software from one computer environment to another. Portability requirements can be specified and designed into the software. Use of a Higher Order Language (HOL) enhances portability. Use of a hardware configuration which is part of a compatible family (i.e., IBM 360/370, PDP-11) enhances portability within that family. It may be unnecessarily expensive to include portability requirements when transfer to another configuration is not envisioned.

Readability is a feature of the code which allows the programmer to quickly identify the portion of interest to him and to easily understand its design. Readability should be proposed by the development contractor in the CPDP.

Reliability is the ability of the software to operate without error*. Reliability is a difficult and perhaps inappropriate term when applied to software because this term has an entirely different meaning for hardware. Since a computer program never wears out it is virtually impossible to predict or analyze failure rates. Any failure of the computer program is a latent design deficiency and its occurrence cannot be adequately predicted. In this respect a computer program cannot be designed for reliability and cannot be tested or evaluated for reliability. Reliability should not apply to computer programs as end items although the computer programs may be used to enhance system reliability.

Self-Containedness is a feature of a module or CPC which allows it to perform all its functions within itself. It should not be specified but can be proposed by the development contractor in the CPDP.

Self-Descriptiveness is a feature of the code and its comments which enables a programmer to understand its structure, its processing flow, and its design intent.

Software Quality is a composite measure of all the software quality characteristics. Although metrics for software quality measurement are currently under development and evaluation, the state-of-the-art for determining software quality is primarily through subjective evaluation.

Structuredness is a feature of the design and code which indicates a pattern of organization of its independent parts.

Testability refers to the ability of the design and code to support evaluation of its performance. In general, well stated performance requirements will result in testable software.

Understandability is a characteristic of the design and code that makes its purpose and functions easy to learn and follow. It is specified through programming standards which include such features as program commenting requirements, naming conventions, limited control structures, and use of structured HOLs.

*In most cases software cannot be verified to be error free (i.e., free from design deficiencies). Under testing and during operations, software errors are uncovered when the software performs in a manner contrary to its performance requirements.

ACRONYMS AND ABBREVIATIONS

AFR - Air Force Regulation

AFSC - Air Force Systems Command

AFSCP - Air Force Systems Command Pamphlet

CAF - Critical Assessment Factor

C^3 - Command, Control and Communications

CDR - Critical Design Review

CDRL - Contract Data Requirements List

CI - Configuration Item

CPC - Computer Program Component

CPCI - Computer Program Configuration Item

CPDP - Computer Program Development Plan

CPT&E - Computer Program Test and Evaluation

CRISP - Computer Resources Integrated Support Plan

CRMC - Computer Resources Management Center

CRWG - Computer Resources Working Group

CSEP - Computer Systems Evaluation Panel

DCP - Decision Coordination Paper

DID - Data Item Description

DT&E - Development Test and Evaluation

ECP - Engineering Change Proposal

ESDM - Electronic Systems Division Manual

FCA - Functional Configuration Audit

FQT - Formal Qualification Test

GFP - Government Furnished Property

IT&E - Independent Test and Evaluation

IV&V - Independent Verification and Validation

PCA - Physical Configuration Audit

PD - Program Director

PDR - Preliminary Design Review

PM - Program Manager

PMD - Program Management Directive

PMP - Program Management Plan

PO - Program Office

PQT - Preliminary Qualification Test

PR - Purchase Request

PSL - Program Support Library

QA - Quality Assurance

QQPRI - Qualitative and Quantitative Personnel Requirements Information

RFP - Request for Proposal

ROC - Required Operational Capability

RSS - Regulations, Specifications, and Standards

SD - Software Director

SDR - System Design Review

SEMP - System Engineering Management Plan

SE/TD - System Engineering/Technical Direction

SOW - Statement of Work

SRR - System Requirements Review

SS - System Specification

SSAC - Source Selection Advisory Council

SSEB - Source Selection Evaluation Board

TBD - To Be Determined

TEMP - Test and Evaluation Master Plan

WBS - Work Breakdown Structure

APPENDIX C - BIBLIOGRAPHY

MILITARY SPECIFICATIONS AND STANDARDS

DoD Directive 5000.29; "Management of Computer Resources in Major Defense Systems;" DoD, 26 April 1976.

MIL-STD-483(USAF); "Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs;" DoD; 31 December 1970.

MIL-STD-490; "Specifications Practices;" DoD, 30 October 1968.

MIL-STD-499A(USAF); "Engineering Management;" DoD; 1 May 1974.

MIL-STD-1521A(USAF); "Technical Reviews and Audits for Systems, Equipment, and Computer Programs," DoD, 1 June 1976.

MIL-C-45662A; "Calibration System Requirements;" DoD; 9 February 1962.

MIL-S-52779(AD); "Software Quality Assurance Program Requirements;" DoD, 5 April 1974.

MIL-Q-9858A; "Quality Program Requirements;" DoD; 16 December 1963.

AIR FORCE AND SUBORDINATE COMMAND DOCUMENTS

AFR 74-1, "Air Force Quality Assurance Program;" USAF; 11 September 1972.

AFR 74-18; "Quality Assurance Program During Acquisition;" USAF(LGPM); 8 November 1976.

AFR 800-14; "Acquisition Management;" Volume II, Acquisition and Support Procedures for Computer Resources in Systems; USAF; 26 September 1975.

AFSCM/AFLCM 375-7; "Configuration Management for Systems, Equipment, Munitions, and Computer Programs;" USAF (AFSC/AFLC); 31 March 1971.

AFSCR 74-6; "Procurement Quality Assurance for System Programs;" USAF (AFSC); 14 July 1973.

"AFSCF Software Engineering and Management Plan;" Headquarters AFSC/XRF; March 1976.

AFSCP 800-3; "A Guide for Program Management;" USAF(AFSC); 9 April 1976.

AFSCP 800-6; "Statement of Work Preparation;" USAF(AFSC); 18 August 1972.

ESDM 74-1; "Quality Assurance for Electronic Systems;" USAF(ESD); 25 February 1974.

RADC-TR-74-300: "Structured Programming Series;" Volume VI, Programming Support Library Program Specifications; USAF(RADC); November 1972.

RADC-TR-74-325; "Engineering of Quality Software Systems;" USAF(RADC):

- Volume V; Measuring the Complexity of Computer Software; Sullivan, J.E.; Prepared by the MITRE Corporation; January 1975.
- Volume VIII; Software Reliability Modeling and Measurement Techniques; La Padula, L.J.; Prepared by the MITRE Corporation; January 1975.

SAMSO Pamphlet 74-2; "Contractor Software Quality Assurance Evaluation Guide;" 1 September 1976.

SAMSO-TR-68-268; "Definition and Evaluation of Merit in Spaceborne Software;" Ruby, R.J., Hartwick, R.D., Nielsen, W.C., Tabler, O.F.; SAMSO; Prepared by Logicon, Inc.; June 1968.

MCI-75-002; "Automatic Data Processing Documents Review Guide;" USAF(ESD); 1 September 1976.

DATA ITEM DESCRIPTIONS

DI-E-3101; "System Specification"

DI-E-3118; "Minutes of Formal Reviews, Inspections, and Audits"

DI-E-3119A; "Computer Program Development Specification"

DI-E-3120A; "Computer Program Product Specification"

DI-E-3121; "Version Description Document (Computer Program)"

DI-E-3128; "Engineering Change Proposals (ECPs)"

DI-E-3129; "Request for Deviations/Waivers"

DI-H-3253; "Qualitative and Quantitative Personnel Requirements Information (QQPRI), Part I - Field and Organization"

DI-H-3267; "Evaluation Needs/Exercise Requirements Analysis"

DI-H-3268A; "Human Operator Task Analysis for Information Systems"

DI-H-3269A; "Training Needs/Exercise Requirements Analysis"

DI-H-3270A; "Exercise Capability Implementation Plan"

DI-M-3409; "Positional Handbook-Information System Operational Personnel"

DI-M-3410; "User Manual (Computer Program)"

DI-S-3618; "System Engineering Management Plan (SEMP)"

DI-T-3703; "Category I Test Plan/Procedures (Computer Program)"

DI-T-3706; "Category II Test Plan/Procedures"

DI-T-3717; "Category I Test Report (Computer Program)"

DI-T-3719; "Category II Test Reports"

GENERAL REFERENCES

"A Defense View of Software Engineering;" Whitaker, Lt. Col. W.A. (USAF); Proceedings of the 2nd International Conference on Software Engineering; Pgs. 358-362; ACM/IEEE/Nat. Bureau of Standards; IEEE Catalog No. 76CH1125-40; October 1976.

"A Design Methodology for Reliable Software Systems;" Liskov, B. H.; Proceedings of Fall Joint Computer Conference; Vol. 41, Part 1; Pgs. 191-199; AFIPS; 1972.

"An Introspective Analysis of DoD Weapon System Software Management;" De Roze, B. C. (Directorate for Weapons Support Systems Acquisition); Defense Management Journal; Pgs. 2-7; October 1975.

"Computer Programming Management;" Auerbach Information Management Series; Auerbach Publishers, Inc.; April 1976.

"Designing Reliable Software;" Ogdin, J. L.; Datamation; Pgs. 71-78; July 1972.

"Effective Software Management Requires Consideration of Many Factors;" Zempolich, B. A. (Asst. Tech. Administrator, Research and Technology Group, Naval Air Systems Command); Defense Management Journal; Pgs. 8-12; October 1975.

"Embedded Computer System Software Reliability;" Manley, Lt. Col. J. H. (USAF) - Asst. for Processor and Software Planning, DCS/Development Plans, Hqs. AFSC; Defense Management Journal; Pgs. 13-18; October 1975.

"Factors in Software Quality;" McCall, J. A., Richards, P. K., Walters, G. F.; Interim Technical Report No. 2 for Contract F030602-76-C-0417; Prepared for ESD/RADC by General Electric Company, Space Division, January 1977.

"Improving the Programming Process;" Cammack, W. B., Rogers, H. J., Jr.; TR 00.2483; IBM; October 1973.

"Increased Software Transferability Depending Upon Standardization Efforts;" Cooper, Cdr. J. D. (USN) - Tactical Digital Systems Office, Naval Material Command; Defense Management Journal; Pgs. 19-22; October 1975.

"Independent Test and Evaluation Guidelines;" Biche', P. W., Morgan, D. J., Wattenberg, R. E.; Logicon Report No. DS-R74036; Prepared for SAMTEC (Contract No. F04701-73-C-0133, Task 74-4) by Logicon; July 1974.

"Managing Software: It Can Be Done;" Romberg, B.W.; Infosystems; Pgs 40-43; November 1973.

"Preview: Blueprint for Software Success;" Manley, J. H. (The Johns Hopkins University Applied Physics Laboratory); Digest of Papers, COMPCON 76; Pg. 38; IEEE; IEEE Catalog No. 76CH1115-56; Library of Congress Catalog Card No. 68-1628; September 1976.

"Problems in Developing The Advanced Logistics Systems;" Report to the Congress by the Comptroller General of the United States; LCD-75-101; GAO; 17 June 1976.

"Quantitative Analysis of Software Reliability;" Shoomun, M. L., et at; IEEE Symposium on Software Reliability; January 1972.

"Quantitative Evaluation of Software Quality;" Boehm, B. W., Brown, J. R., Lipow, M.; Proceedings of the 2nd International Conference on Software Engineering; Pgs. 592-605; ACM/IEEE/Nat. Bureau of Standards; IEEE Catalog No. 76CH1125-4C; October 1976.

"Quantitative Measurement of Program Quality;" Rubey, R. J., Hartwick, R. D.; Proceedings of 23rd ACM National Conference; Pgs. 671-677; ACM; 1968.

"Reliability of Real-Time Systems;" Yourdon, E.; Modern Data; Serialized (six parts); January-June 1972.

"Software Development;" Mills, H. D.; Supplement to Proceedings of 2nd International Conference on Software Engineering; Pgs. 79-86; ACM/IEEE/Nat. Bureau of Standards; IEEE Catalog No. 76CH1125-4C; October 1976.

"The Influence of Software Structure on Reliability;" Parnas, D. L." International Conference on Reliable Software; April 1975.

"The Mythical Man-Month;" Brooks, F. P., Jr.; Datamation; Pgs. 45-53; December 1974.

"Why Projects Fail;" Keider, S. P.; Datamation; Pgs. 53-55; December 1974.

COMMENT SHEET

Software Quality Assurance Guidebook

Reviewers Name: Reviewer's Organization:

Comments:

Please return to: Hq ESD/MCIT (Stop 36)
Hanscom AFB, MA 01731